Towards Generic Domain-specific Information Retrieval

Zhao Jin

B. Comp. (Hons.), NUS

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2013

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Min-Yen Kan. Without his guidance, patience and support over all these years, this thesis would not have been possible.

I would also like to express my gratitude to other established researchers for their comments and research opportunities at different stages of my Ph.D. They are Prof. Yin Leng Theng, Prof. Paula M. Procter and Prof. Tamara Sumner.

Thanks also go to my colleagues and friends in the Computational Linguistic Lab and the Web Information Retrieval / Natural Language Processing Group (WING), especially Long Qiu, Hendra Setiawan, Shanheng Zhao, Yee-Fan Tan, Zhi Zhong, Jesse Prabawa Gozali, Ziheng Lin, Jun Ping Ng, Pi-Dong Wang, Xuancong Wang, Aobo Wang, Tao Chen and Xiangnan He. I certainly had a lot of great times discussing with them about research, life and many other topics. They have made my Ph.D. years much more enjoyable.

Last but not least, I can never thank my family and friendmily too much for their love and care. I am very blessed to have them in my life.

Contents

Abstract

To improve domain-specific information retrieval, we have identified and examined two generic (domain-independent) but prominent problems in this area: **Resource Categorization** and **Text-to-Construct Linking**.

The first problem refers to the categorization of domain-specific resources at multiple granularities. This helps a search engine to better meet specific user needs by highlighting task-relevant materials and organize its presentation of search results by more pertinent metadata criteria.

The second problem refers to the resolution of domain-specific concepts to their related domain-specific constructs. This allows constructs to properly influence relevance ranking in search results, without troubling users to input them in potentially awkward construct syntax.

We observe correlations among various characteristics of domain-specific resources, capturing them in a multi-layered graph. Following this graph, we carry out our research on the two aforementioned problems as follows: For Resource Categorization, we use the key information extraction problem in healthcare as a case study on the categorization of correlated nominal facets. We exploit the correlation between two categorizations at different granularities (*i.e.*, sentence-level and word-level) by propagating information from one to the other sequentially or simultaneously. In addition, we use the readability measurement problem as a case study on the categorization of ordinal facets. We exploit the correlation between the readability of domain-specific resources and the difficulty of domain-specific concepts through iterative computation. For Text-to-Construct Linking, we tackle the linking of math concepts to their representations in math expressions. We exploit the correlation between the observable characteristics of

a concept-expression pair and its relation type using supervised learning.

To demonstrate the applicability and usefulness of our research, we have implemented two domain-specific search systems, one in the domain of math and the other in healthcare. Both systems incorporate and extend our research findings to handle domain-specific user needs. Our evaluation shows that both the Resource Categorization and the Text-to-Construct Linking features are effective in facilitating domain-specific search.

List of Tables

List of Figures

Chapter 1

Introduction

As digital libraries and resources proliferate, how scholars find, access and use information changes. Researchers, teachers, students and the general public increasingly turn to online search engines for quick, indicative searches and even for longer sessions of information gathering. Such searches often begin as general keyword searches to large, publicly-available search engines.

However, such a search strategy works poorly for domain-specific information retrieval (IR). Based on our preliminary user study of math search [Zhao et al., 2008] and subsequent research, there are two key issues associated with this search strategy in the context of finding relevant domain-specific resources:

First, users feel that general search engine results are disorganized. Different types of resources in the results are mixed together without internal organization. Many scholarly disciplines have a wide range of resources on the Web, where topics are explained using different modes: a brief definition from a dictionary page, a tutorial with examples and exercises, or a research paper with rigid proofs. Each of these modes caters to different audiences, ranging from neophytes to research specialists. In the domain of math, the topic of modular arithmetic serves as a case in point: Simple examples can be explained to children in the guise of clock arithmetic, but specialists' needs in ring theory might start with searches composed of identical keywords but are in fact looking for papers to keep themselves abreast of cutting-edge research progress. As another example, in the healthcare domain, registered practice nurses need information about a disease or a healthcare practice of interest, whereas research nurses need to find studies that validate certain healthcare practices for particular diseases. However, few

general search engines are able to recognize such modes and organize the results accordingly. As a result, users must expend a lot of effort navigating through the results to find the ones aligned to their needs.

Moreover, users also feel that there is a lack of support for applying selection criteria on the search results in general search engines. In domain-specific IR, users often have in mind a set of selection criteria that help to decide which resources are the most suitable. Such criteria are mostly concerned with desirable characteristics of the resources. The stronger those characteristics are in the resources, the more likely they will be selected by the users. For example, due to the technical nature of medical knowledge, articles in the medical domain are often too specialized for the general public [Graber et al., 1999]. Therefore, laymen prefer more readable articles, thus making readability one of the most important selection criteria to be supported in medical search. Likewise, when educators search for teaching resources, they apply multiple selection criteria, such as the prestige of the sponsors, appropriateness for the target students' age range, and the degree of organization, to ensure that the selected resources are of high quality. However, the automatic measurement of these characteristics, which is the prerequisite for providing such support, is still in its early stage (with the exception of readability). Therefore, the application of these selection criteria is likely to remain a manual and time-consuming process for users. How to automate this process is a challenge for researchers.

Second, while it is desirable to make domain-specific constructs searchable and relevant in ranking, users still prefer to use text keywords over other input modalities. Many scholarly disciplines have their own domain-specific constructs to encode information. These constructs convey precise, detailed information about knowledge in a domain. Examples include DNA sequences, molecular formulas, music notation, and, in the domain of math, mathematical expressions. These domain-specific constructs lead to two difficulties in current search technology. First, although they are comparatively better than natural language in terms of compactness, expressiveness, and operative power, construct notation is far more difficult to analyze and utilize in retrieval. For example, despite the fact that a large amount of information is encoded as math expressions in math-

ematical documents, math expressions are seldom a factor in relevance ranking. Second, inputting constructs can be troublesome and awkward. Even if we assume that the first difficulty is solved, users hoping to use construct-aware search may have a difficult time entering constructs to form queries. For example, in math search, on-screen keyboards and equation editors can be used to construct a math expression, but these are still at best awkward to use. Considering the fact that math expressions are still mostly text-based, this problem is exacerbated in other domains where constructs also have a non-textual component (*e.g.* molecular structures in chemistry or modern music notation).

These two issues surface in many domains and need to be addressed in the corresponding domain-specific search engines. However, instead of treating these problems with domain knowledge (which we believe is fruitful and many times, necessary), in this thesis, we work towards finding suitable approaches to address these problems **without** domain knowledge. We aim to further approaches for domain-specific IR in a general, domain-independent manner – *i.e.*, not requiring expensive domain knowledge sources such as ontologies and knowledge bases – so that the techniques can be ported to any domain easily. In this way, we can improve domain-specific IR in general instead of only in a few specific domains.

We believe that the first issue can be addressed by **Resource Categorization**, *i.e.*, the automatic categorization of resources on both nominal (*e.g.*, resource type) and ordinal (*e.g.*, readability) facets. If automated, this categorization would enable search engines to organize results for easier navigation and provide better support for the application of selection criteria. For example, a search on "modular arithmetic" will return several smaller lists of results, one for each mode of resources, with options to rank the results in each list by relevance, readability or quality. Novices can then filter out materials other than readable tutorials, while researchers can route their interests directly to research papers.

In order to address the second issue, we examine a related yet somewhat different problem: **Text-to-Construct Linking**, *i.e.*, to link domain-specific concepts together with domain-specific constructs, so that the constructs relevant to concepts can be identified, analyzed and utilized as part of ranking. For example, a search on "Pythagorean theorem" would be recognized as equivalent

to a search on $x^2 + y^2 = z^2$ and resources containing this or other construct variants would also be marked as relevant.

Upon close inspection, we have observed that both problems involve determining certain characteristics associated with domain-specific resources at different granularities. For example, in Resource Categorization, the key characteristics can be larger, resource-level characteristics, such as resource type and readability, as well as more fine-grained sentence- or word-level characteristics, such as sentence or word type. As for Text-to-Construct Linking, the key characteristics can be the relation type between a concept and a construct in a sentence. Correlations exist among these characteristics, which can be exploited in solving the aforementioned problems. For example, knowing the type of a sentence may help to infer the word types within the sentence, and vice versa. We represent these characteristics and correlations in a graph and use it to guide the problem solving process for these problems.

Based on this graph, we exploit the following correlations using domain-independent approaches to address the problems of Resource Categorization and Text-to-Construct Linking:

- For Resource Categorization on nominal facets, we exploit the correlation between two categorizations at different granularities (*i.e.*, sentence- and word-level) by propagating information from one to the other, sequentially or simultaneously.

- For Resource Categorization on ordinal facets, we measure the readability of domain-specific resources. To exploit its correlation with the difficulty of domain-specific concepts, we use an iterative computation algorithm to recursively estimate one from the other.

- For Text-to-Construct Linking, we link domain-specific concepts to their related constructs using supervised learning. The correlation exploited in this problem is the one between the observable characteristics of a concept-construct pair and its relation type.

In the subsequent sections, we will detail our correlation graph, describe the goals and contributions of our research, and outline the structure of this thesis.

1.1 Correlation Graph for Domain-specific Resources

Given our dissection of the two major tasks needed in catering to domain-specific IR, what approaches are appropriate to address them? *Ad hoc* methodologies can be applied to each specific domain but such methods would not capitalize on the shared structures that we believe exist across different domains.

A methodology that has been used in wide variety of tasks to model structure is graphical representation. Any characteristics and correlations can be naturally represented as nodes and edges in a graph. Suitable computational mechanisms can then be employed to exploit specific correlations as a way to determine the characteristics of interest based on others. As such, we also capture the characteristics of domain-specific resources and their correlations in a graph.

We define *domain-specific resources* as textual resources written for certain domain-specific concepts in styles suitable for their purposes. They are one of the most common targets of retrieval in domain-specific IR.

Although commonly retrieved as individual resources, they can also be viewed as a hierarchy of segments. We define *segments* as parts which the resources are divided into based on certain criteria. For example, when the resources are first divided into sentences and then words, the resources can be viewed as a hierarchy of two levels with sentences being the segments at the first level and words being the segments at the second level.

Various characteristics can be associated with domain-specific resources, specifically to the concepts for which the resources are written, the resources themselves as a whole and the segments in the resources. As a few examples, the concepts for which the resources are written can be associated with *difficulty*, which measures the amount of prerequisite knowledge required to understand a concept. The resources themselves as a whole can be associated with *resource type*, which is the genre of a resource defined based on the types of information it contains and how such information is organized, *readability*, which measures how difficult it is to understand a resource, and *average sentence length*, which is the average number of words per sentence in a resource. The segments in the resources can be associated with *segment type*, which we define as the type

of information a segment contains or represents, and *relation type*, which is the type of semantic relation that exists between two segments.

Many of these characteristics are *correlated* in the sense that knowing one of the characteristics will help to infer another. For example, knowing the type of a domain-specific resource helps to infer the types of the segments it contains and vice versa, while knowing the readability of a resource can help to infer the difficulty of the concepts it is written for and vice versa. Such correlations are useful when we need to infer certain characteristics based on others.

The resulting graphical representation of such characteristics and correlations is our *correlation graph*. It can be used to guide the research on many problems in domain-specific IR pertaining to the indexing and retrieval of domain-specific resources, including Resource Categorization and Text-to-Construct Linking.

We now go through the topology of our graph and describe its application for problem solving in domain-specific IR.

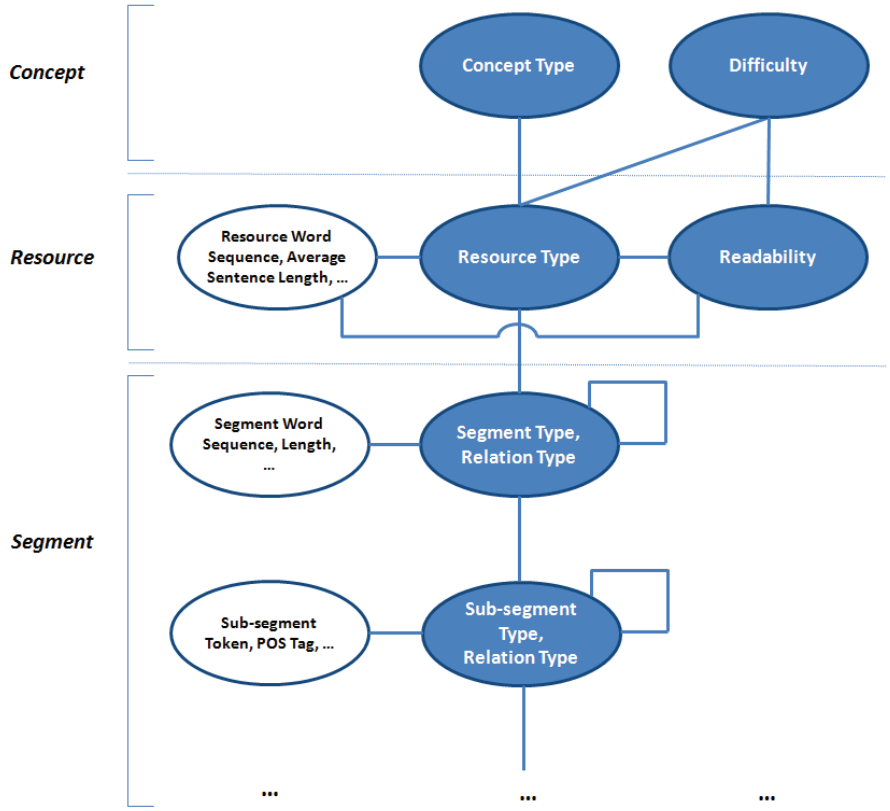
1.1.1 Topology

We propose a topology of our correlation graph for domain-specific resources, shown in the example in Figure 1.1. In this graph, the nodes in white represent observable characteristics associated with domain-specific resources, such as word sequence and average sentence length, while the ones in grey represent hidden characteristics, such as resource type and readability. These nodes take on one or more values whose types and meanings vary depending on the characteristics they are representing. For example, the values for the node representing resource type can be nominal categories, such as tutorials and papers, while the values for the node representing readability can be ordinal ranks, such as grade levels. Edges are undirected, representing correlations among the characteristics.

The graph itself is divided into three layers: concept, resource and segment, each representing a different aspect of domain-specific resources.

The *concept layer* represents the domain-specific concepts for which a resource is written. The nodes in this layer represent characteristics such as difficulty and concept type. For example, in terms of difficulty, addition and subtraction are easy since they can be learned with little math knowledge, whereas

Figure 1.1: Example correlation graph for domain-specific resources. The nodes represent characteristics associated with domain-specific resources. The colors of the nodes (*i.e.*, white or grey) indicate whether the corresponding characteristics are observable or not. The edges are undirected and represent the correlations between pairs of characteristics.



integration and differentiation are more difficult because they require a more comprehensive domain background. As another example, in terms of type, Fourier transform and Pythagorean theorem are examples of operation concepts and theorem concepts, in math respectively. Likewise, diabetes and vitamin are examples of disease concepts and substance concepts, in medicine respectively. Since the focus of our graph is on domain-specific resources, we keep this layer simple and do not model possible correlations among the characteristics of the concepts. Therefore, there are no edges among the nodes in this layer.

The *resource layer* represents a domain-specific resource as a whole. The nodes in this layer represent characteristics such as resource type, readability, and average sentence length. These nodes are correlated with each other as indicated by the edges among them. For example, the average sentence length node is correlated with the readability node since average sentence length is

indicative of the readability of a resource.

The *segment layer* represents the segments in a domain-specific resource. Depending on the segmentation granularit(ies), this layer may contain multiple levels. Each level corresponds to a different granularity. The levels collectively form a hierarchy of segments. The nodes in each level represent characteristics such as segment type, relation type, and word sequence in a segment. There may also be correlations among the nodes within or across the levels in this layer. For example, the word sequence in a sentence is indicative of its type (*e.g.*, example sentences usually start with the phrase "For example"). In the medical domain, the type of a sentence may give evidence for specific word types (*e.g.*, a sentence describing the patients of a medical study is likely to contain words that represent patient demographics).

The three layers in our graph do not exist in isolation. Rather, there are many correlations among the characteristics from different layers. For example, difficulty in the concept layer is correlated with readability in the resource layer, as resources written for difficult concepts are generally less readable, while concepts commonly described by less readable resources are more likely to be difficult. As another example, between the resource and the segment layers, resource type and segment type are correlated. Knowing the resource type helps to determine the possible segment types in a resource (*e.g.*, a course website usually contains information about textbooks on the concepts to be covered in a course) and vice versa (*e.g.*, a resource with plenty of definitions and examples of concepts is more likely to be a tutorial than a resource hub).

The nodes, edges and layers as described above form our correlation graph for domain-specific resources. For more detailed lists of example nodes and edges in the graph, please refer to Appendix A.1.

1.1.2 Problem Solving with Correlation Graph

In our opinion, a fundamental problem in domain-specific IR is to facilitate the information seeking process of domain-specific searchers by characterizing domain-specific resources in the presence of domain-specific concepts and constructs, without relying on expensive domain knowledge sources.

There are several reasons why we pose this as a fundamental problem:

First of all, IR of any type should aim to assist users in their information seeking process. Domain-specific IR is no exception to this. Given the complexity of domain-specific searchers, search systems that support these domains would not work well without first understanding their needs and then catering to them.

Second, the characteristics of domain-specific resources are crucial in facilitating the domain-specific information seeking process. For example, characteristics of the resources as a whole, such as resource type and readability, allow supporting search systems to retrieve more relevant results and assist users in determining suitable resources from such results more easily. As another example, characteristics that may serve as domain knowledge (*e.g.*, the relation types between domain-specific concepts and constructs) can be utilized in ranking or presented to users directly to satisfy their information needs. Therefore, it is important to determine such characteristics in domain-specific IR.

Lastly, although domain knowledge sources make it easier to utilize domain knowledge, they are costly to compile and their availabilities vary from domain to domain. Hence, we cannot rely on them in niche or underresourced domains.

The two problems examined in our research (*i.e.*, Resource Categorization and Text-to-Construct Linking) are both instances of this fundamental problem:

The problem of Resource Categorization is to categorize resources on various facets (*i.e.*, characteristics of interest) at multiple granularities, such as resource type, readability, sentence type and word type. It facilitates the information seeking process by allowing search engines to organize results better and enabling users to navigate through search results (*e.g.*, filtering by resource type and sorting by readability) to select suitable ones (*e.g.*, checking whether the study design described in a research article is valid) more easily.

The problem of Text-to-Construct Linking is to semantically relate domain-specific concepts to constructs. It facilitates the information seeking process in different ways, depending on the nature of the semantic relations of interest (*e.g.*, connecting concepts with their construct representations saves users' trouble of inputting the constructs manually). The characteristic of interest in this problem is the relation type of a pair of concept and construct.

Table 1.1: Examples of Resource Categorization.

Name	Problem Description
Genre Classification	To categorize resources based on the information they contain and how such information is organized.
Information Extraction	To categorize segments (*e.g.*, sentences/words) of resources based on the information they contain/represent.
Concept/Construct Recognition	To identify whether a word/symbol is part of a domain-specific concept/construct.
Metric Measurement	To measure the readability/specificity/cohesion of resources.

Table 1.2: Examples of Text-to-Construct Linking.

Name	Problem Description
Representation Identification	To identify representations of domain-specific concepts in constructs.
Operand Role Labeling	To label the roles of constructs with respect to the operations (represented by domain-specific concepts) applied on them.
Co-reference Resolution	To find the constructs referred to by domain-specific concepts.

More examples of these problems can be found in Table 1.1 and 1.2.

A *correlation graph* can serve as a guide in solving these problems. Given the characteristics of interest, the first step is to identify from the graph a set of nodes that represent such characteristics. New nodes can be added in appropriate layers as necessary. For example, to represent the specificity of a resource, a node can be added in the resource layer.

The second step is to identify from the graph a set of edges that represent the correlations to be exploited in determining the characteristics of interest. This can be done by using the existing edges as a reference and/or performing a corpus study. New edges can also be added among appropriate nodes as necessary. For example, similar to readability, specificity should be correlated to the observable characteristics and the resource type in the resource layer, as well as some hidden ordinal characteristics in the concept layer. A corpus study on domain-specific resources with simple correlation metrics, such as Pearson's R, may reveal that it is correlated with concept genericity (*i.e.*, resources written for more generic concepts are usually less specific) and hence edges can be added between the

corresponding nodes in the respective layers.

Once the set of relevant nodes and edges has been decided, we select an appropriate computational mechanism based on the nature of the characteristics and correlations represented by the nodes and edges. Our correlation graph does not impose a choice of computational mechanisms; we are free to choose a means best suited to the characteristics of interest.

Take the problem of Resource Categorization as an example. We differentiate the two cases where the facets to be categorized are nominal or ordinal. For the former, we examine the categorization of two correlated nominal facets: sentence type and word type. As represented in Figure 1.2, these two facets are correlated to each other in sense that the type of a sentence determines the possible word types in that sentence while the types of the words in a sentence serve as strong indicators of the sentence type. Therefore, we have applied supervised learning for this problem and compared various ways of combining the two categorizations together so that one could inform and improve the other. For the latter, we examine the problem of readability measurement. As represented in Figure 1.3, the readability of domain-specific resources is correlated to the difficulty of domain-specific concepts, since readable resources are commonly written for easy concepts, while difficult concepts are commonly described by less readable resources. To exploit this correlation, we iteratively compute the readability of domain-specific resources based on the difficulty of domain-specific concepts and vice versa.

Figure 1.2: Example set of nodes and edges for Resource Categorization on nominal facets.

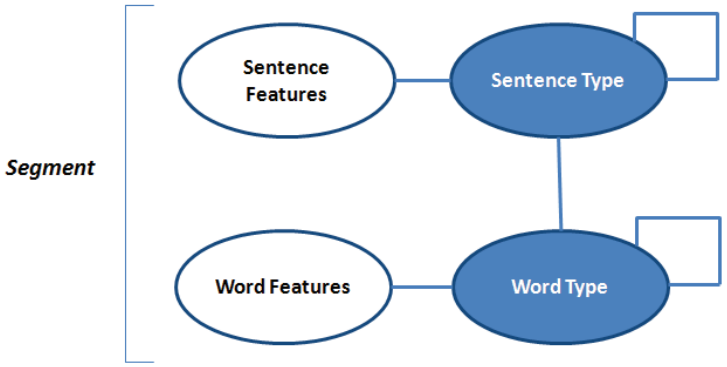
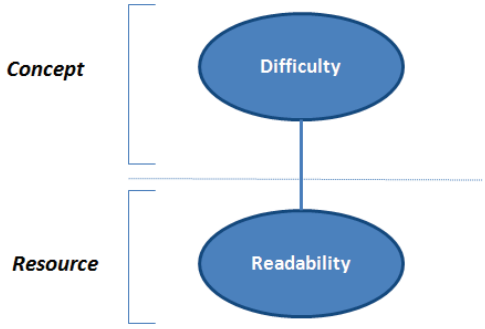
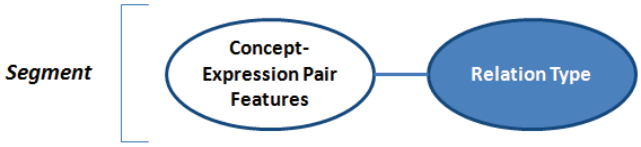


Figure 1.3: Example set of nodes and edges for Resource Categorization on ordinal facets.



As another example, for Text-to-Construct Linking, we are interested in relating math concepts to their representations in expressions. Therefore, the relation type between a concept and an expression is the center of attention in this problem. As represented in Figure 1.4, relation type is correlated with the observable characteristics of a pair of concept and expression. Since relation type is also nominal, our approach is also based on supervised learning as we have done for the first case of Resource Categorization.

Figure 1.4: Example set of nodes and edges for Text-to-Construct Linking.



1.2 Goals and Contributions

Our research aims to improve domain-specific IR in general without using expensive domain knowledge sources. Within this broad aim, we achieve the following three specific goals:

1. To identify prominent problems in domain-specific IR. These problems should be sufficiently common yet addressing them should facilitate domain-specific IR.

2. To address the identified problems in a generic manner so that different instances of such problems in different domains can be addressed similarly.

3. To incorporate the research findings into domain-specific search systems. This helps to verify the usefulness of our research and improve domain-specific IR in practice.

We have made the following contributions towards these goals:

- **Identifying two prominent problems in domain-specific IR.** We identify Resource Categorization and Text-to-Construct Linking as two prominent problems in domain-specific IR based on our user study. These two problems are prevalent in many domains and shall be addressed to aid the resource selection process and alleviate the need for construct input.

- **Providing domain-independent approaches to address the two prominent problems.** We have observed correlations among various characteristics of domain-specific resources and captured such information in a multi-layered graph. Following this graph, we examine the problems of Resource Categorization and Text-to-Construct Linking. By using concrete instances of these problems as case studies, we demonstrate that Resource Categorization may benefit from 1) propagating information between two correlated classifications of nominal facets at different granularities, and 2) iteratively computing the values of two correlated ordinal facets based on each other. To address Text-to-Construct Linking, one possible soution is to first detect the links between pairs of domain-specific concepts and constructs, and then rank the constructs linked to the same concept heuristically to find the suitable ones for display and retrieval. None of these approaches rely on expensive domain knowledge sources and hence they are largely domain-independent.

- **Implementing two domain-specific search systems.** To demonstrate the applicability and usefulness of our research, we have also implemented two domain-specific search systems, one for math and the other for healthcare, based on our research findings. These systems may serve as platforms for domain-specific IR research and can be expanded into practical systems for public use in future.

1.3 Thesis Outline

The rest of this thesis is organized as follows.

In Chapter 2, we give an overview of the research in domain-specific IR, detail the user study from which we identify the two problems examined in our research, and review existing works on how graphical representations have been applied in general and domain-specific IR.

In Chapter 3, we examine Resource Categorization on nominal facets. In particular, we compare several ways to exploit the correlation between categorizations at different granularities. This is done through a case study on the problem of key information extraction in healthcare.

In Chapter 4, we continue our investigation in Resource Categorization but shift our focus to ordinal facets. Using readability measurement for domain-specific resources as a case study, we demonstrate that an iterative computation algorithm can be employed to exploit the correlation between two ordinal facets for better measurement accuracy.

In Chapter 5, we move on to the problem of Text-to-Construct Linking. We approach this problem by a two-step process consisting of concept linking and construct ranking. We carry out this part of research in math, linking concepts to their expression representations.

In Chapter 6, we introduce the math and healthcare search systems we have built. Both systems have incorporated features based on our research on Resource Categorization and Text-to-Construct Linking.

In Chapter 7, we conclude this thesis. We first recap the contributions of our research and then point out possible directions for future research.

Chapter 2

Background

We start our related work survey by reviewing domain-specific IR research. We then detail our user study from which we derive the two primary problems for this thesis' focus. As we use a graphical perspective to find the commonalities in domain-specific IR, in the end, we review the relevant literature on graphical representations and related work that motivates our correlation graph for domain-specific resources. We defer the reviews specific to the individual research problems to their respective chapters.

2.1 Domain-specific IR

Domain-specific IR is a type of vertical search that focuses on a specific domain. The term 'domain' here refers to a particular sphere of knowledge, influence, or activity. Common examples of domains include (but are not limited to) general sciences, such as math, medicine and bio-informatics, and humanities, such as law, economics and music.

The main objective of domain-specific IR is to obtain domain knowledge and/or resources that can be used to appreciate, learn or apply domain knowledge. It overlaps somewhat with other types of vertical search when the resources of interest are of particular media types (*e.g.*, text webpage and videos) or genres (*e.g.*, tutorial and research paper); however, in domain-specific IR, the domain knowledge in the resources should be the primary concern. For example, a search for movies can be considered as domain-specific IR if the intention is to appreciate the domain knowledge (*e.g.*, cinematic techniques) in the movies; however, if the search is just to obtain movies for personal enjoyment, it is not considered

as domain-specific IR because in this case, the domain knowledge contained in movies is not the primary focus.

There are several key elements that need to be taken into consideration in domain-specific IR:

The first element is the presence of domain knowledge. We define *domain knowledge* as the facts and information in a particular domain. It is referred to by domain-specific concepts, encoded by domain-specific constructs, described in domain-specific resources and captured in domain knowledge sources. Such knowledge is also possessed and sought after by domain-specific searchers.

The second element is the presence of domain-specific concepts. We define *domain-specific concepts* as the natural language phrases used to refer to pieces of domain knowledge. For example, "operator" is a biological concept that refers to a segment of DNA, while "ring theory" is a math concept that refers to the study on a particular type of algebraic structures. It is important to be able to recognize them from domain-specific resources and handle them specifically for retrieval instead of treating them as normal text phrases. For example, a search engine for biological information should recognize "operator" as a domain-specific concept from a research article and know that it is related to the concept "DNA". When the concept "DNA" is used as a query, the domain-aware search engine can then use this piece of information to infer that this article may be relevant, too, even though it may not mention "DNA" explicitly. As another example, a math search engine needs to recognize that "ring theory" is a difficult concept even though it is a combination of two simple words, and that the presence of this concept will decrease the readability of a resource.

The third element is the presence of domain-specific constructs. We define *domain-specific constructs* as the symbolic representations which encode domain knowledge through a domain-specific way other than natural language. For example, math expressions are domain-specific constructs in math since they represent math knowledge through combinations of symbols such as numbers, variables and operators. As another example, songs can be considered as domain-specific constructs in music when interpreted as an arrangement of notes of varying pitches, timbre and rhythm. These constructs need to be handled with

specialized indexing and searching techniques so that they can be utilized in retrieval or even become the targets of retrieval themselves. For example, a math search engine needs to be able to analyze the expression $a^2 + b^2 = c^2$ syntactically and semantically to know that it is in the form of "the sum of squares of two variables equals the square of another variable" and is a representation of "Pythagorean theorem". The resources that contain this expression can then be returned when users search for expressions of the same form or resources about Pythagorean theorem. Similarly, a music search engine may analyze a song to know that it is in the style of jazz and return it in response to a search for examples of jazz music. Note that domain-specific constructs are symbolic and independent of how they are stored. For example, the expression $a^2 + b^2 = c^2$ can be stored as a LaTeX expression or an image while songs can be stored as mp3 or midi files, without affecting the knowledge encoded.

The fourth element is the presence of domain-specific resources. As defined in Chapter 1, *domain-specific resources* are textual resources (*e.g.*, a scholarly article, a webpage, a formalized educational lesson module and a newspaper clipping) written for certain domain-specific concepts (*e.g.*, modular arithmetic in math, bird flu in medicine and proteins in bio-informatics) in styles suitable for their purposes (*e.g.*, an introductory tutorial for beginners and a journal information page for researchers). They are the targets of retrieval in most domain-specific searches and domain-specific concepts and constructs frequently appear in them as means to refer to and encode domain knowledge, respectively.

The fifth element is the presence of domain knowledge sources. We define *domain knowledge sources* as domain knowledge compiled in an explicit way that can be utilized directly. Examples of domain knowledge sources include ontologies, which list the concepts in a domain and indicate the relationships among them, and knowledge bases, which use sets of rules to describe domain knowledge in a logically consistent manner. They commonly serve as sources of information which domain-specific search systems can tap on as they handle domain-specific resources. For example, domain-specific search systems can make use of ontologies to recognize concepts from resources and decide whether to return a particular resource based on whether the concepts it contains are semantically related

to the ones in the query. These sources can be very detailed and capture rich nuances of domain knowledge (Element 1), and can be expensive to build and invest in. For example, in medical domain, the UMLS Metathesaurus[1] is a large, multi-purpose, and multi-lingual thesaurus that contains millions of biomedical and health related concepts, their synonymous names, and their relationships. It was released more than ten years ago and is now still being updated twice a year by the National Library of Medicine (NLM) under government support. To be clear, in our thesis, we focus on investigating how to improve domain-specific IR generically, without utilizing these resources, as their availabilities vary from domain to domain.

The last key element is the presence of domain-specific searchers. We define *domain-specific searchers* as the people who seek for domain-specific resources and constructs, as well as the underlying domain knowledge. Their needs are more specialized than general searchers, as they have different roles and exhibit a wide spectrum of domain knowledge. For example, the needs and behaviours of a primary school student will be quite different from the ones of a seasoned researcher, although they may both start their search with the same keyword "modular arithmetic". The student may only need some simple animations illustrating what modular arithmetic is, but ends up being overwhelmed by the mixed results returned and cannot decide which results to pursue in more detail. On the other hand, the researcher, with a stronger background in the domain, is able to differentiate which results are likely to be relevant. He may even reformulate the query using domain knowledge or switch to specialized search engines as necessary. Given the domain as context, it becomes feasible and important to analyze these user needs and behaviors and cater for them specifically.

These key elements interact and pose challenges in domain-specific IR.

For each domain, there will be specific retrieval needs that condition on the specialized knowledge of the domain. Handling these intricacies is not the focus of this thesis. Instead we focus on addressing the common problem patterns that re-occur in many domains.

Based on our literature review on IR in specific domains, such as math,

[1] http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

medicine and music, we have noted three major challenges in domain-specific IR: 1) indexing and searching domain-specific resources, 2) indexing and searching domain-specific constructs, and 3) query languages.

2.1.1 Indexing and Searching Domain-specific Resources

The indexing and searching of domain-specific resources is a major challenge in domain-specific IR, due to the key elements involved.

Approaches for handling domain-specific concepts in domain-specific resources commonly start with the identification of such concepts from the resources. The domain knowledge sources involved could be lexica, thesaurii or ontologies which list the concepts in a domain and possibly encode the relationships among them. By taking into account the presence of such concepts in the resources and the relationships among them as derived from the domain knowledge sources, the retrieval process can then replace standard keyword search with concept-based search, or augment standard searching techniques with the help of such concept information. For example, [Meij et al., 2009] investigate language models based on concepts instead of words for domain-specific IR, while [Hliaoutakis et al., 2006] enhance the standard vector space model by introducing concept semantic similarity scores derived from MeSH (Medical Subject Heading[2]) in medical domain. A few other works, such as [Kim and Compton, 2001] and [Radhouani et al., 2009], also explore organizing the resources according to concept ontologies to allow for easier navigation to resources of related concepts.

Dealing with domain-specific constructs in domain-specific resources is more tricky. It involves a number of tasks including identification, analysis, storage and matching of constructs. To be more specific, first, the constructs need to be identified from the resources. Afterwards, they are analyzed both syntactically and semantically and then converted into suitable internal representations. In the end, these internal representations are matched with the queries from users during retrieval. All of them are non-trivial and domain-specific issues, such as the nature of constructs (*i.e.*, to deal with constructs with complex structures) and notational variation (*i.e.*, to determine whether two seemingly different con-

[2] http://www.nlm.nih.gov/mesh/

structs are equivalent), make them even more challenging.

Taking the domain of math as an example, the identification of math expressions is done by symbol recognition and structural analysis based on supervised learning [Chan and Yeung, 2000]. The remaining three tasks are solved collectively and the common approaches can be text-based or non-text-based. Text-based approaches treat the math expressions as text and apply standard IR techniques for both searching and indexing. Searching can be as simple as token matching (*e.g.*, MathWorld[3] and Zentralbatt Math[4]) or pattern matching [Kohlhase and Sucan, 2006]. Lucene, a high-performance text retrieval library, is also deployed for more sophisticated indexing and searching capability [Miner and Munavalli, 2007]. On the other hand, MIaS (Math Indexer and Searcher) [Sojka and Líška, 2011] and MathWebSearch [Kohlhase et al., 2012] are two examples of non-text-based approaches. The former employs unification algorithms to create more generalized versions of the expressions while the latter parses the expressions into substitution trees (more commonly used in symbolic math systems, such as theorem provers). Both methods abstract away the surface symbols and hence are able to overcome the notational variation problem. Similar research efforts can also be seen in other domains such as chemistry. ChemxSeer [Mitra et al., 2007] indexes not only the chemical formula but also the tables in chemistry resources so that they become searchable in the system.

In addition, categorization – the characterization of resources by type, organization, intended audience or other dimensions – is also necessary so that suitable resources can be selected to meet the needs of domain-specific searchers. In general IR, this is commonly done in the guise of genre classification and readability measurement. Nevertheless, the complex needs of domain-specific searchers and the presence of domain-specific concepts also increase the complexity of categorization. For example, [Price et al., 2007; Price et al., 2009] show that, besides genre, identifying the semantic components, *i.e.*, "segments of text about a particular aspect of the main topic of the document and may not correspond to structural elements in the document", helps the retrieval of

[3] http://mathworld.wolfram.com/
[4] http://www.zentralblatt-math.org/zmath/en/

health-related documents. As another example, [Yan et al., 2006] show that readability measurement in domain-specific IR can be improved by taking into account the scope of domain-specific concepts and their semantic relationships.

2.1.2 Indexing and Searching Domain-specific Constructs

In the domains where the domain-specific constructs are sufficiently complex, they can become the targets of retrieval themselves. Songs in music IR serve as a case in point, where users may want to search for songs from a music library to learn more about music. The indexing and retrieval of such constructs can be done based on their contents and/or additional information annotated on them.

Content-based approaches extract a feature vector/matrix for each construct, match it with the one from the query to obtain a similarity score and then perform ranking. The actual features extracted depend heavily on the nature of the domain-specific constructs and vary from domain to domain.

Take music IR as an example, there can be low-level features, such as signal parameters, Mel-Frequency Cepstral Coefficients (MFCCs), and psychoacoustic information [McKinney and Breebaart, 2003], as well as high-level ones, such as pitch [Zhu et al., 2001], timber [Scaringella, 2008] and rhythm [Foote et al., 2002]. The computation of similarity can be as simple as distance measures [Logan and Salomon, 2001] but advanced statistical techniques (*e.g.*, Independent Component Analysis [Pohle et al., 2006] and Mean-Covariance Restricted Boltzmann Machine [Schlüter and Osendorfer, 2011]) are not uncommon. Similarly, in artwork IR, [Zirnhelt and Breckon, 2007] use weighted k-Nearest Neighbour to retrieve artworks based on color and texture features, while [Jiang et al., 2004] extract non-objectionable semantics, such as warmth, contrast and saturation, to allow users to query on such semantics explicitly.

If the constructs are annotated with information such as name, source and description, retrieval can leverage them to supplement knowledge gleaned from the constructs' content. For example, text-based retrieval methods can be applied on metadata when users are able to specify their queries with suitable vocabu-

laries (*e.g.*, searching in Allmusic[5] for songs and Artcyclopedia[6] for artworks). As another example, with the rapid growth of social networks, recommendation systems based on collaborative filtering (*e.g.*, getting recommendations for songs from last.fm[7] and for movies from Rotten Tomatoes[8]) have also become an excellent alternative for content-based retrieval systems.

In addition, it is also possible to categorize domain-specific constructs for retrieval. For example, in music, two common facets for categorization are genre (*e.g.*, rock/jazz/hip-hop) [Scaringella et al., 2006] and mood (*e.g.*, happiness/anger/sadness) [Feng et al., 2003], while in photography, photos can be categorized by scene (*e.g.*, indoor/outdoor and manmade/natural) [Boutell and Luo, 2005]. In general, machine learning approaches are prevalent [Scaringella et al., 2006; Bosch et al., 2007] for this purpose.

2.1.3 Query Languages

Since domain-specific constructs are not based on natural language, it is a challenge in domain-specific IR to find a query language which is expressive enough to specify the constructs for domain experts, yet accessible to lay users.

For domain-specific constructs that are largely text-based, such as math expressions and chemical formula, many types of solutions are available. The simplest way is to write them in plain text (*e.g.*, a2+b2=c2 and C2H4). This is highly accessible but not very expressive. In contrast, specialized languages, such as LaTeX[9] (general-purpose), MathML[10] (math) and CML[11] (chemistry), are very expressive yet much less accessible due to their steep learning curves. Lastly, graphical user interfaces (*e.g.*, onscreen equation editors) are somewhere in between in the sense that they allow lay users to write complex constructs using a predefined (usually limited) set of symbols and operators.

For domain-specific constructs that are not text-based, query by example is

[5] http://www.allmusic.com/
[6] http://www.artcyclopedia.com/
[7] http://www.last.fm
[8] http://www.rottentomatoes.com/
[9] http://www.latex-project.org/
[10] http://www.w3.org/Math/
[11] http://www.xml-cml.org/

a popular approach (*e.g.*, searching for songs by humming using Midomi[12] and finding visually similar photos using Google Image[13]).

2.2 User Study in Math

While we are able to identify the challenges in domain-specific IR through a literature review, it is unclear to us what the desiderata for domain-specific search systems are and whether the current research adequately satisfies these desiderata. To better understand the desiderata and formulate our research problems, we have conducted a user study in the domain of math.

Given this objective, we believe it is important to observe users' actual seeking process *in situ* and allow for more exploratory and productive tangential discussions to take place immediately. Therefore, we choose to use a qualitative, semi-structured interview rather than a quantitative survey instrument. Therefore, the results we report here are necessarily preliminary and indicative, but are descriptive and allow us to posit and justify our system design (to be detailed in Chapter 6). Similar study design has been used by [Bishop, 1998], among others. Using this format, we have interviewed 13 volunteer participants including 2 undergraduates, 7 graduate students, 1 professor and 3 librarians, all affiliated with the math department of NUS.

We have a checklist of topics (and associated probe questions) for discussion during interviews. Except for the ones on simple demographics (*e.g.* their experience in searching for math resources), our questions loosely correspond to the various stages of the Big6 Information Seeking Model [Eisenberg and Berkowitz, 1990]. These include what kind of resources they typically look for (Task Definition), how they approach searching (Information Seeking Strategies), what resource collections they use (Location and Access), as well as their expectations for a math search system (Evaluation).

We interviewed the subjects in their typical work environment so that we could observe their natural seeking behaviors. After first introducing the goals of our research and disclosing the interview conditions, we conducted the inter-

[12] http://www.midomi.com/
[13] http://www.google.com/imghp

view according to our checklist. Participants were encouraged to discuss other pertinent issues and demonstrate their seeking behaviors on a math topic of their choice. On average the interviews lasted 30 minutes and were not recorded; however, summary notes were compiled during each interview. After each interview, we open-coded the summary notes and consolidated our findings. We continued interviewing and recruiting new participants while new findings were uncovered. Our findings stabilized after ten interviews, so we concluded the study after a final round of three more interviews.

2.2.1 Key Findings

Although there are many findings from our user study, in this subsection, we choose to review only three of them which directly connect to the desiderata. They are, namely, keyword search, mathematical expression input and user needs. For more details, please refer to our earlier work [Zhao et al., 2008].

Keyword Search

With regards to their own information seeking process, participants have reported that they commonly search the Web using a *general search engine* querying for math concepts. Compared to other information seeking approaches, such as browsing and personal contacts, this approach is very popular because of its short response time and high availability, as well as the variety of resources it provides. On the other hand, the participants have complained about its inaccuracy and the lack of organization in the results. Such problems often drive them to switch from general search engines to media-specific (*e.g.*, Google Books[14]) or domain-specific (*e.g.*, MathWorld) ones. When pressed about how organization may be improved, it is clear that standard IR topical clustering is not sought; but clustering by purpose, by resource type or by audience level.

Mathematical Expression Input

As identified in our literature review on domain-specific IR, input and retrieval of domain-specific constructs (*i.e.*, math expressions in this case) is a focal point

[14] http://books.google.com/

of current efforts. Although our participants have expressed general interest in such facilities, when probed for specific applications, surprisingly, most are unable to picture a scenario where expression search may be useful. The only potential usage mentioned by an undergraduate is to find problem set solutions.

All other participants have doubts in the value of such facilities, either due to the lack of mathematical expressions in their research domain, the inconvenience of entering expressions, or the high specificity of math expressions.

When asked to hypothesize about how they would prefer to input math expressions, all participants have stated that they would prefer to input in LaTeX. This is tied to familiarity, as it is the math expression authoring tool of choice.

These negative findings in our survey indicate that the current domain-specific IR research focus may not really address the basic problems encountered by users, and that a cognitive gap exists between users and researchers.

User Needs

What types of resources are our participants looking for? From our post-analysis, we observe that all queries involved math concepts, and requirements on its content or style (*i.e.*, format). We characterize these needs into two broad categories: *Information needs* center on content (*e.g.*, definition of complex numbers) while *resource needs* seek resources in a particular format (*e.g.*, articles on set theory). This is similar to the observations in web query analysis [Broder, 2002]. Table 2.1 gives a complete list of the identified needs.

Table 2.1: Types of math user needs identified.

Information	Name, definition, derivation, explanation, example, problem, solution, graph, chart, algorithm, application and related concept.
Resource	Paper, tutorial, slides, course website, book, code, toolkit and data.

By factoring together commonalities in our participants' comments, two other (usually tacit and unstated) facets of user needs have also emerged in helping them to select relevant resources. *Readability* measures how difficult it is to understand a resource. If a resource is too hard for users to understand, it is not

helpful however relevant it is. *Specificity* measures the level of details at which the concepts are discussed in a resource. Less specific resources are sufficient for a general, indicative understanding of the target concepts while more specific ones give a thorough, informative understanding of the mathematical basis of the concepts. These two facets are often correlated but distinct.

2.2.2 Desiderata in Domain-specific IR

Given the evidence from our interviews, we feel that there is an unmet need for a math search engine. Such a system should address user needs more directly without additional burdens to the users.

Is the current work in math IR able to fill these gaps? Unfortunately, we do not find this to be the case. According to the participants in our study, natural user-driven applications of the current math IR work may be limited, even in cases where expert users (professors and graduate students) are concerned. Moreover, current research efforts center around math expressions: their input (as queries), indexing and retrieval. From our study, it is clear that users find text input the most viable form of searching and specialized input modalities for equations unwieldy. With this in mind, we identify two problems which we feel domain-specific search systems should address: Resource Categorization and Text-to-Construct Linking.

- **Resource Categorization:** Our study find that the participants feel the general search engine results are disorganized and different types of resources which are logically separate are presented together. This is not specific to math. In almost any domain, there are various types of resources written for the same concept with different purposes and audiences. For example, for the same concept, a webpage may explain it with animations for children, a tutorial may define it concretely and provide exercises to help students learn it, a paper may address a research problem related to it, while a resource hub may list down all the above as resources that are related to it. All these may be returned in response to a keyword search on the concept and lead to the organization problem as observed in math[15].

[15]Similar concerns have been voiced out by the healthcare practitioners in the development

Therefore, we believe a key need in domain-specific search is automatic Resource Categorization. A domain-specific search engine must classify resources automatically, ensuring that different needs requiring different types of information or resources are satisfied, without distracting irrelevant search results. From our study, we believe that automatic classification on facets such as readability is also helpful to narrow down relevant resources. Such automatic faceted classification results need to be integrated using a suitable, faceted searching/browsing user interface so that the results can be organized as needed to facilitate resource selection.

- **Text-to-Construct Linking:** Domain-specific search engines will be more compelling if they are domain-aware and able to leverage the domain-specific constructs in a useful way. However, through our user requirements study, we conclude that the usability of such search methods is a problem: General users find keyword search most effective and do not feel that inputting equation is easy. While expert users may be satisfied with specialized construct authoring languages, the general audience of math IR engines would not find them accessible due to their steep learning curves. Given the fact that domain-specific constructs are not written in natural language, we believe similar usability problems also exist in other domains since it usually takes more time and effort to learn how to formulate queries with constructs and apply it during actual searches than using keywords in natural language. Nevertheless, we believe this does not suggest that construct retrieval is irrelevant; rather, the question is how we could make the search and ranking of constructs relevant to users while maintaining the usability of keyword search.

  We believe a method to bridge this usability gap lies in automatically relating domain-specific concepts and constructs. We propose that Text-to-Construct Linking, *i.e.* the resolution of concepts to the related constructs (*e.g.*, Pythagorean theorem to $a^2+b^2=c^2$), will work as a form to retrieve

process of our healthcare search system. They are interested in finding full text research articles that verify the effectiveness of a medical intervention on certain patients; however, many other resources, such as webpages that explain it in plain words for laymen and textbooks that explain its procedures in detail for students, are returned in the search results in a disorganized manner.

constructs relevant to a concept. The constructs retrieved this way can be presented to users as information and used to retrieve domain-specific resources that contain similar constructs. All these can be done without requiring users to input such constructs explicitly.

2.3 Graphical Representation

As mentioned in Chapter 1, both Resource Categorization and Text-to-Construct Linking aim to determine the characteristics of domain-specific resources at different granularities by exploiting their correlations. Therefore, a suitable representation for domain-specific resources should be able to naturally represent such characteristics and correlations.

The simple bag-of-words model used in traditional IR does not meet this requirement. It represents resources as unordered collections of words. Therefore, it is unable to model them beyond word-level (*e.g.*, unsuitable in representing individual sentences). Moreover, since it disregards grammar and word order, the context of words – useful in understanding information they represent – is also lost. Therefore, it is not suitable for representing domain-specific resources. Similarly, although the vector space model and the language model are more expressive and capture more information (*i.e.*, the importance of words through term weighting and the language properties of resources as probability distributions, respectively) than the bag-of-words model, they are still largely limited to capturing word-level information. Therefore, they are not suitable either.

As we look for better representations for domain-specific resources, graphical representations emerge as a suitable choice because the characteristics and correlations can be naturally represented as nodes and edges in a graph.

2.3.1 Common Graphical Representations

A graphical representation is a graph structure containing nodes representing elements to be modeled, and edges representing the relationships between them. Given a collection of entities (*e.g.*, resources and queries), a graph can be constructed and a suitable computational mechanism can be applied on the con-

structed graph to derive the information of interest. For example, researchers can be modeled as a graph in which nodes represent the researchers themselves while the undirected edges among the nodes represent the fact that they have co-authored papers. Similarly, webpages can be modeled as nodes with directed edges from one to another representing the fact that the former contains a link to the latter.

In these general graphical representations, the information of interest can be captured as structural patterns or scores. Following the earlier example of co-authorship graph, as is done in [Merlin and Persson, 1996], patterns such as "most researchers have only a few coauthors, while a few have very many hundreds or even thousands in some cases" and "biological scientists tend to have significantly more coauthors than mathematicians or physicists" can be recognized. On the other hand, as is done in the HITS algorithm [Kleinberg, 1999], a hub score and an authority score can be assigned to a node in webpage graph to represent the value of the content of a webpage and the value of its links to other webpages. These two scores can be iteratively computed as the sum of all the authority scores of the nodes it points to and the sum of all the hub scores of the nodes that point to it. These graphical representations have been widely studied in the context of social network analysis [Carrington et al., 2005], biological network analysis [Junker and Schreiber, 2008] and link analysis [Thelwall, 2004].

Moreover, graphical representations admit a probabilistic interpretation when their nodes represent random variables while their edges encode not only relationships but also conditional independence between nodes. For example, Bayesian networks [Pearl, 1985] have directed edges which are often used to (but not required to) represent the casual relationships between nodes (*i.e.*, an edge from node A to B denotes that A causes/influences B). A node in a Bayesian network is conditionally independent of any other nodes given its Markov blanket which consists of its parents (*i.e.*, the nodes which have an edge pointing this node), children (*i.e.*, the nodes which are pointed to by an edge from this node) and the children's parents. In contrast, Markov networks [Kindermann and Snell, 1980] have undirected edges representing the dependencies between nodes (*i.e.*,

an edge between node A and B denotes that A and B are mutually dependent). A node in a Markov network is also conditionally independent of any other nodes given its Markov blanket which in this case, consists of its neighbours.

In these probabilistic graphical representations, the information of interest is encoded as the joint distribution of all the nodes in the network. Since both Bayesian and Markov networks encode conditional independence, this joint distribution can be decomposed into a product form of probability distributions (*i.e.*, the conditional probabilities in Bayesian networks and the potential functions in Markov networks). Therefore, they are highly compact representations of the joint probability table. This advantage has made them very popular in many different domains, such as bioinformatics [Friedman et al., 2000; Wei and Li, 2007], medicine [Mani et al., 2005; Descombes et al., 1998] and decision making [Jensen and Nielsen, 2007; Bhattacharjya et al., 2009].

In short, graphical representations provide a simple and sound framework for representing elements and their relationships in any domain. When coupled with suitable computational mechanisms, they can serve as tools for reasoning/computation as well.

2.3.2 Graphical Representations in General IR

In general IR, specifically web searches, general graphical representations are often used to derive information about webpages based on hyperlinks. Besides the HITS algorithm mentioned earlier, PageRank [Page et al., 1998] and SALSA [Lempel and Moran, 2000] are two other well-known link analysis algorithms. The former determines the importance of a webpage based on the intuition that the number of backlinks of a webpage is a good indication of its popularity or importance, while the latter combines the strength of PageRank and HITS by incorporating the backlink information into the hub and authority computation. Despite the success such algorithms have achieved, the graphical representations behind them are only at resource- (*i.e.*, webpage-) level and hence not detailed enough for modeling domain-specific resources.

In contrast, probabilistic graphical representations have more to offer when it comes to modeling the resources in detail. Bayesian networks made their debut

in IR in the 1990s [Turtle and Croft, 1991; Fung and Favero, 1995] as a modeling tool for the retrieval process. Such networks often consist of two levels: one level for documents and the other for queries. While the direction of the edges differs among works, all such representations basically model the relationships between document and query features. Retrieval is then to rank the documents according to their posterior probability of relevance given the query. These basically lay the groundwork for IR using a Bayesian network methodology. Subsequently, work has been done to further enhance the modeling of documents and queries: For example, [Metzler and Croft, 2004] combine Bayesian networks with a language model to allow for a rich, structured query language; the series of works by de Campos and his colleagues [de Campos et al., 2000; Crestani et al., 2003; de Campos et al., 2004] model the dependencies among the query terms and the structural units of the documents by linking together the respective nodes and forming them into subnetworks. [Tsikrika and Lalmas, 2004] also examine the impact of hyperlink-based evidence on retrieval effectiveness when combined with other content-based evidence.

In comparison, the introduction of Markov networks to IR occurred much later. [Metzler and Croft, 2005] describe an IR model based on a two-level Markov Random Field, one for query terms with several possible dependency models (*i.e.*, independent, sequential and full) and the other for documents with dependencies to each of the query terms. Potential functions between query terms can then be defined in a way similar to language models while the ones between the documents and the query terms can be defined based on a variety of textual and non-textual features. The notion of relevance in this framework is the joint probability of the nodes in the graph having the values representing the documents and the query terms. This model was extended by later works to better handle queries [Metzler and Croft, 2007; Lease, 2009]. A notable extension of this model is to introduce another layer of nodes representing the topical segments of the documents, as is done in [Lang et al., 2010] for the purpose of query expansion. Along a similar line of thinking, in image retrieval, [Feng and Manmatha, 2008] construct a Markov network with images represented as a set of visual terms which are linked to individual query terms. This model was

extended in [Llorente et al., 2010]. In their model, dependencies between terms are modeled and the images are represented by any visual features instead of just visual terms.

We note that the work to incorporate the documents' structural information into graphical representations thus far has been limited and has the potential to be improved. So far only generic document structures such as sections and paragraphs [Crestani et al., 2003] or topical segments [Lang et al., 2010] are considered. Neither is suitable for domain-specific resources since the unit of information in the resources do not necessarily conform the generic document structure (*e.g.*, the definition of a domain-specific concept may span over a few sentences within a paragraph) while segmentation done by other (*e.g.*, functional and visual) criteria other than topics can be useful, too.

2.3.3 Graphical Representations in Domain-specific IR

The application of general graphical representations in domain-specific IR up to the current date has been focused on citation analysis. Certain domain-specific resources, such as papers, books and journals, are connected through citations and hence can be translated into citation graphs. Based on these graphs, metrics can be computed to measure the importance of domain-specific resources and influence the ranking process. For example, the number of citations a paper receives may serve as a quick indication of the importance of the paper, while the impact factor [Reuters, 2012] measures the importance of journals as the average number of citations received per paper published in that journal during the two preceding years. Nevertheless, as is the case in general IR, these graphical representations seldom go beyond resource-level.

As for probabilistic graphical representations, Bayesian networks have been applied mainly to combine multiple pieces of evidences and model the uncertainties in the retrieval process. As a few examples, [Schuller et al., 2003] use Bayesian networks to integrate multimodal queries and contextual knowledge for music retrieval. [Silveira and Ribeiro-Neto, 2004] use them to consolidate the concept-based rankings which are generated by matching the related concepts of the query to the ones in judical documents, while [Quellec et al., 2008; Quellec

et al., 2011] use them to handle various sources of information, which might be incomplete, uncertain and conflicting, for medical experts in diagnosis. On the other hand, Markov networks are much less widely applied in domain-specific IR. To our knowledge, [Yu et al., 2009] is the only work that applies Markov networks in domain-specific IR. They examine transfer learning with Markov networks to transfer useful prior knowledge from an existing dataset to a new dataset for better retrieval performance. This should be useful for adapting domain-specific search systems to new domains.

Nevertheless, little work has been done to explore how to model and utilize the structure of domain-specific resources with graphical representations in domain-specific IR. As far as we know, only the later works by de Campos [de Campos et al., 2006; de Campos et al., 2008] apply Bayesian networks to retrieve domain-specific resources (*i.e.*, medical records and parliamentary documents); however, the representations used in these works are still the generic ones.

2.3.4 Insights from other Areas

To get a better understanding of how resource modeling can be done, we have looked towards other areas to find representations of domain-specific search and resource structure. The user study we have described earlier has informed us that domain-specific user needs center around both content (*i.e.*, type of information) as well as format (*i.e.*, how the information is organized). Therefore, we believe that there is a need to label segments of a resource according to the type of information presented and the resource itself according to its format. Correspondingly, the resource representation should further model resource segments in addition to itself, as is confirmed by [Price et al., 2007; Price et al., 2009].

Relevant work in examining and categorizing fragments of webpages exists in the area of information extraction. We note that several of these works (*e.g.*, [Schapke and Scherer, 2004; Wong et al., 2008]) use a layered probabilistic network that models the generation of a webpage fragment starting from the conceptual entity. Both insights have inspired us to come up with our proposed layered graph to combine their strengths to make a generic yet well-structured representation to handle the indexing and searching of domain-specific resources.

2.4 Summary

Despite the fact that the research on IR with graphical representations has started two decades ago, the document representation in these works did not go beyond word- (concept-) level and was constrained by the generic document structure. We believe this is a major limitation, as the findings from our user study indicate that certain user needs require segments more fine-grained than the document as a whole, but more coarse-grained than just the word-level. By looking at works from other areas, we have confirmed our belief that the resource representation should further model segments in addition to itself and noted that relevant works in information extraction use a layered probabilistic framework to model the generation of segments starting from conceptual entities. To draw on the successes of these works, we have proposed to also use a layered graph in modeling domain-specific resources as described in Section 1.1.

Chapter 3

Resource Categorization on Nominal Facets – A Case Study in Key Information Extraction for Evidence-based Practice

As pointed out in Chapter 2, domain-specific search engines should be able to categorize domain-specific resources automatically so that specific user needs can be satisfied by specific types of resources without distracting irrelevant results.

This problem of Resource Categorization is a broad topic in the sense that it can be done at many different granularities and on many different facets. For example, at the top level, resources can be categorized by resource type (*i.e.*, the genre of a resource defined based on the types of information it contains and how such information is organized) and readability (*i.e.*, how difficult it is to understand a resource). At the middle level, segments or sentences that compose the resources can be categorized based on the types of information they contain (*e.g.*, definitions, examples and proofs). At the bottom level, words and domain-specific constructs can be categorized according to the types of information they represent (*e.g.*, person names, locations and patient demographics) and their forms (*e.g.*, math variables/operators, chemical elements/compounds and DNA codes/sequences) respectively.

To make our investigation into this problem more manageable, we have divided it into two sub-problems: one for nominal facets and the other for ordinal facets. The values of nominal facets are categories which are distinct from

each other. A common way to handle these facets is to treat their values as separate classes and apply supervised learning to perform the desired categorization [Sebastiani, 2002]. At more fine-grained levels (*i.e.*, sentence-level and below), rule-based extraction is also popular [Sarawagi, 2008]. In contrast, the values of ordinal facets are meant to establish an ordering. Therefore, traditional approaches for such facets simply compute some scores heuristically (*e.g.*, the Flesch-Kincaid Reading Ease formula [Flesch, 1948]) to derive the ordering. Although it is possible to treat them as nominal by using an ordered set of categories as values, the fact that they are relative and inexact in nature calls for a different way of handling. Therefore, we focus on nominal facets in this chapter and save the discussion on ordinal facets for the next chapter.

Resource categorization on nominal facets has been studied in various contexts but often only at one specific granularity level.

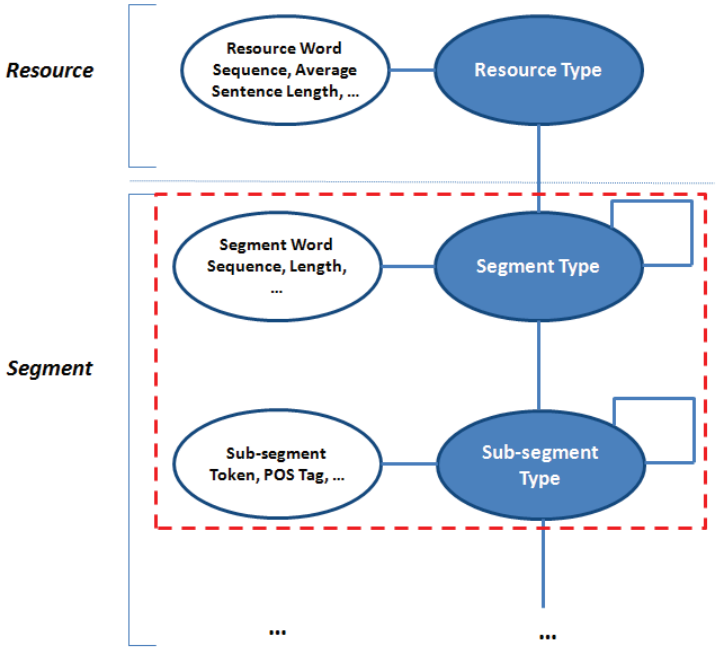
For example, genre classification [Lee and Myaeng, 2002] is performed at resource-level, while named entity recognition [Nadeau and Sekine, 2007] and bio-informatics information extraction [Tanabe and Wilbur, 2002] are at word-level. Although classification at sentence-level is often employed in the context of question answering [Demner-Fushman and Lin, 2007] and information extraction [Sitter and Daelemans, 2003] to identify the sentences that contain the information of interest, it is treated as a preprocessing step instead of part of the main task in such contexts.

We believe this is a limitation in domain-specific IR, due to two reasons. First, the categorization of domain-specific resources needs to be done at multiple granularities. Only in this way would users be able to filter out unsuitable results by coarse-grained facets and then select the most appropriate ones based on fine-grained facets. In addition, more coarse-grained categorizations may serve as a fallback when more fine-grained ones are unreliable or unable to capture the desired information well. Second, without considering categorizations at different granularities simultaneously, their correlations, which are often useful in improving categorization performance, will be left unexploited.

Therefore, we choose to focus on investigating how to improve categorizations of different granularities by exploiting the correlations among them. To this

end, we use key information extraction for evidence-based practice in healthcare as a case in point. The problem of *key information extraction* is to extract information pertinent to evidence-based practice, such as patient demographics, interventions, results and study design, from research articles in the form of sentences and words. As such, there are two correlated categorizations to be performed: one at sentence-level and the other at word-level.

Figure 3.1: Correlation graph fragment showing nodes and edges relevant to segment and sub-segment type. The edges (*i.e.*, correlations) bounded by the dashed line box are examined in this chapter.



In our correlation graph (Figure 3.1), to categorize at these two levels is to find the values for two nodes: segment (*i.e.*, sentence) type and sub-segment (*i.e.*, word) type in the segment layer. As represented by the edges in the graph, these two nodes are correlated with many other nodes including those above in the resource layer or below the sub-segment level. Since our primary interest is to examine how the categorizations of two different levels interact, we ignore the correlations beyond these two levels. This leaves us six correlations as bounded by the dashed line box shown in Figure 3.1. Without considering the correlation between the two categorizations (*i.e.*, treating the two categorization as independent), the remaining correlations simply mean that the categorization of a segment can be done based on the information from itself and its context as

established by the types of its neighbours. This is suboptimal, as knowing the segment type helps to determine the sub-segment type and vice versa.

For example, in key information extraction, knowing that a sentence describes patients in a medical study will increase the likelihood that the words in this sentence represent patient demographics (*e.g.*, age and sex) and vice versa. Therefore, for this part of our research, we have treated these categorizations as supervised classification problems and examined how to exploit their correlation through propagating information between the classifications.

We believe the findings from our research can be applied to improve resource categorization on other pairs of correlated nominal facets, such as resource type and segment type. We will elaborate on this towards the end of the chapter.

The rest of the chapter is organized as follows. We start with a detailed description of the problem of key information extraction for evidence-based practice in the domain of healthcare in Section 3.1, followed by a literature review in Section 3.2 on entity extraction and key information extraction. Then we present our models for exploiting the correlation between categorizations for key information extraction in Section 3.3. We evaluate the performance of the models with different settings and explore the effects of data filtering and feature selection in Section 3.4. We present directions for future research in Section 3.5 and end with a discussion on Resource Categorization on nominal facets based on our findings in Section 3.6.

3.1 Key Information Extraction for Evidence-based Practice

Evidence-based practice (EBP) is the integration of best research evidence with clinical expertise and patient values [Sackett et al., 2000]. EBP promotes the synthesis and critical appraisal of healthcare literature to meet the information needs of practitioners, and accelerates the adoption of research findings into practice. It has become commonplace in healthcare in recent years.

Despite the growing popularity of EBP in healthcare, support for the gathering and selection of applicable and valid research articles in today's EBP collec-

Table 3.1: Definitions of PICO elements.

Name	Definition
Patient	The description of the patient. It commonly consists of five sub-elements: sex, co-morbidity, race, age and pathology (SCORAP).
Intervention	The intervention applied.
Comparison	Another intervention examined as a comparison or control.
Outcome	The outcome of the experiment.

Table 3.2: PICO elements of a sample clinical question.

Clinical Question: For a 54-year-old woman with periodontal disease, how effective is the therapeutic use of doxcyline decrease gum bleeding and recession compared to no treatment?	
P	54-year-old (age) woman (sex) with periodontal disease (pathology)
I	Doxcyline
C	No treatment
O	Decrease gum bleeding and recession

tions can still be improved. Published guidelines [Sackett et al., 2000] recommend that a clinical question needs to be established using PICO elements [National Health and Medical Research Council, 1999] (*i.e.*, patient, intervention, comparison and outcome) as shown in Table 3.1 and 3.2. These identified elements can serve as the criteria in determining the applicability of a research article.

Beyond the PICO elements, there is also a hierarchy in the strength of evidence [National Health and Medical Research Council, 1999] for articles as shown in Table 3.3. This hierarchy helps a reader to assess the validity of the research articles, as stronger evidence (*i.e.*, articles of a lower grade) is generally preferred.

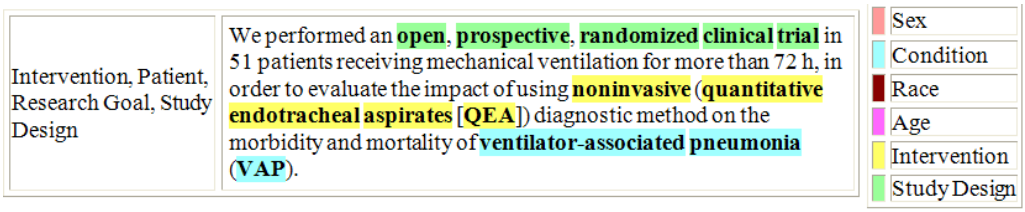
However, common EBP collections seldom provide such information explic-

Table 3.3: Different levels of strength of evidence.

Grade	Definition
I	Systematic reviews of all relevant Randomized Controlled Trials (RCTs)
II	At least one properly designed RCT
III-1	Well designed pseudo-RCT
III-2	Cohort studies, case control studies, interrupted time series without control
III-3	Comparative studies with historical control, two or more single-arm studies or interrupted time series without control
IV	Case series

itly or allow users to filter for these criteria. Although users may be able to perform keyword searches and limit their searches by gender, age and study design in PubMed[1], they cannot specifically target keywords which match only the text sections about PICO elements or strength of evidence. As such, users must resort to reading the abstract or even the full text of an article to figure out whether it is indeed applicable and valid.

Figure 3.2: Display of extraction results to assist the users in applicability and validity assessment.



We believe the extraction of such information from articles is the key to solve this problem. With such information extracted, additional features can be implemented into search systems to support the assessment process. For example, as illustrated in Figure 3.2, a system can display the key sentences of a research article and highlight the keywords that reveal key information such as intervention and study design in those sentences. Users can then assess the applicability and validity of the articles immediately without the need to read them in full. Moreover, this extraction has to be done automatically, since manual extraction would be too labor intensive due to the large amount of research articles available.

3.2 Literature Review

The automatic extraction of structured information from unstructured sources has been an active area of research for more than two decades. According to the taxonomy of information extraction proposed by [Sarawagi, 2008], this research area can be categorized along five dimensions: the type of structure extracted, the type of unstructured source, the type of input resources available for extraction, the method used for extraction and the output of extraction. Under these

[1] http://www.ncbi.nlm.nih.gov/pubmed

dimensions, the problem of key information extraction is to extract and output entities (*i.e.*, words and sentences that help to determine the applicability and validity of an article) from unstructured texts (*i.e.*, research articles). For the rest of this section, we first review the relevant methods and input resources for the extraction tasks of similar nature, and then move on to the work specific to key information extraction.

3.2.1 Entity Extraction from Unstructured Texts

The methods for entity extraction from unstructured texts can be broadly classified into two categories: rule-based and statistical.

Rule-based Approaches: As the name suggests, rule-based approaches rely on a set of rules to perform extraction. Rules usually consist of two parts. The first part is a contextual pattern which describes the properties and context of the entities to be extracted in terms of textual features. As summarized in [Muslea, 1999], early information extraction systems for newspaper articles make use of lexical features (*e.g.*, the words themselves), phrase features (*e.g.*, noun/verb/prepositional groups), voice features (*e.g.*, active/passive) and word type features (*e.g.*, physical object) to construct complex patterns. The second part of the rules is the action to be taken when the patten is matched, which is usually to identify series of words as the entities to be extracted.

These rules can be hand-crafted by experts or learnt from an annotated corpus. Hand-crafted rules are able to encode domain knowledge which is hard to capture otherwise and feature widely in early systems [Hobbs et al., 1997; Cunningham et al., 2002]. To alleviate the cost of domain knowledge, rule-learning algorithms have been developed to induce the best set of rules based on an annotated corpus and rule templates. The learning of rules may start by instantiating very specific rules from the templates to cover instances of the information to be extracted, followed by a generalization process that removes some of the text features or replaces rules with more general ones. This is bottom-up rule learning as is done in [Ciravegna,

2001]. Alternatively, the learning can be done in a top-down manner. In [Soderland, 1999], generic rules are made more specialized by adding more text features or replacing them with more specific ones. Nevertheless, these algorithms may still rely on existing hand-crafted rules as a better starting point and involve experts in instance selection and rule refinement for better results.

Despite the growth of statistical approaches, rule-based approaches remain an active area of research and efforts have been made to improve them in various aspects, such as scalability [Reiss et al., 2008], uncertainty management [Michelakis et al., 2009] and refinement process [Liu et al., 2010].

Statistical Approaches: In statistical approaches, the extraction of entities is done by classifying whether a word is (part of) an entity to be extracted using statistical models. The words in such approaches are commonly described by a set of text features consisting of word features (*e.g.*, the words themselves), orthographical features (*e.g.*, capitalization pattern), linguistic features (*e.g.*, part-of-speech tags) and dictionary features (*e.g.*, whether the word appears in the entity dictionary). Under this formulation, various statistical models have been examined by different researchers. Hidden Markov Models (HMMs), which naturally capture the dependency between adjacent words, feature prominently in early research. For example, [Bikel et al., 1997] use an Ergodic HMM with internal states representing named entity classes. They calculate the most likely state for each word using the Viterbi decoding algorithm. Later works employing HMMs in information extraction focus on finding the suitable model structure [Seymore et al., 1999] or employing more sophisticated variants of HMMs such as Hierarchical HMMs [Skounakis et al., 2003]. Besides HMMs, Support Vector Machines (SVMs) and Maximum Entropy modeling (MaxEnt) have also been applied in [Isozaki and Kazawa, 2002] and [Chieu and Ng, 2002] for their capability in handling large amount of features. [McCallum et al., 2000] propose the Maximum Entropy Markov Model which combines the strength of HMM and MaxEnt in capturing sequential dependency while

offering more freedom in the choice of features. This leads to the current state-of-the-art model, Conditional Random Fields (CRFs) [Lafferty et al., 2001], which is able to take into account larger context (instead of just the previous word) for individual input and construct a consistent sequence of labels as the output. As a more recent trend, efforts have been made to solve multiple related information extraction tasks together via joint inference [McCallum, 2006; Poon and Domingos, 2007] so that the results of one classification can be used to inform another and vice versa.

Both categories of approaches rely on the presence of an annotated corpus, which is often expensive to obtain. To alleviate the tedium and cost of building large corpora, semi-supervised learning [Nadeau, 2007; Carlson et al., 2010] and unsupervised learning [Etzioni et al., 2005; Dalvi et al., 2012] methods have also been studied for various entity extraction tasks.

Our approach for key information extraction is statistical, as such approaches require less domain knowledge as compared to rule-based approaches (where experts are involved in crafting and tuning the rules). This domain independence allows our approach to be applied in different domains without having to source for expensive domain knowledge and makes our findings more applicable to domain-specific IR in general.

3.2.2 Key Information Extraction

In healthcare domain, the identification and utilization of PICO elements and their variants have been studied extensively for various intents. Most of the previous works in this area are based on supervised learning with natural language processing techniques. For example, [Demner-Fushman and Lin, 2007] perform sentence extraction on abstracts to obtain information for clinical question answering. They consider the sentences for elements P, I and C to be more recognizable by patterns due to the presence of medical concepts while the ones for element O to have no predictable patterns. Therefore, they extract the former using hand-crafted patterns but employ linear regression of text features for the latter. [Chung and Coiera, 2007] seek for a better understanding of the structure of clinical abstracts by classifying their individual sentences into five

classes – aim, method, participants, result and conclusion. [Kim et al., 2010] explore the use of lexical, semantic, structural and sequential information with CRFs, while [Boudin et al., 2010] test and combine multiple classifiers, such as Decision Trees, SVM and Naïve-Bayes. Both of these later works improve the accuracy of sentence classification.

In comparison, research on more fine-grained extraction of EBP information is less common. Existing works usually start by classifying the sentences in abstracts or articles to identify the possible locations of EBP information and then proceed to extract the information from those locations. For example, [Bruijn et al., 2008] make use of an SVM-based sentence classifier with n-gram features and a rule-based pattern extractor to identify the key trial design elements from clinical trial publications. [Chung, 2009] extracts interventions from method sentences in RCTs using lexical and syntactical features.

The above works either focus on sentence extraction or use sentence extraction as a basis for keyword extraction. While individually important tasks, we believe that the composition of both tasks together is synergistic and would lessen the effort needed in applicability and validity assessment.

- Sentence extraction is important because not all key information is modeled well by individual words. For example, research results are commonly described in prose. It is difficult to extract only a few words to represent the entire text. Extraction at sentence-level is ideal in this case. Even for information such as patient demographics that can be represented by a few words, sentence extraction still imparts evidence that the specific keywords are being used in an appropriate context.

- Keyword extraction is also important because the recognized keywords represent the exact information users need. With the extracted keywords highlighted based on their classes for the ease of reading and assessment, users may quickly locate the desired information from the sentences without having to go through each of them in detail. Furthermore, keyword extraction aims at a smaller unit of text and hence can be represented in a more compact manner (*e.g.*, keyword clouds) than sentences. This is

Table 3.4: Classes for sentences.

Name	Definition	Example
Patient	A sentence containing information of the patients in a study.	A convenience sample of 24 critically ill, endotracheally intubated children was enrolled before initiation of suctioning and after consent had been obtained.
Result	A sentence containing information about the results of a study.	Large effect sizes were found for reducing PTSD symptom severity (d = −.72), psychological distress (d = −.73) and increasing quality of life (d = −.70).
Intervention	A sentence containing information about the procedures of interest and the ones as the comparison/control in a study.	Children 6 to 35 months of age received 0.25 ml of intramuscular inactivated vaccine, and those 36 to 59 months of age received 0.5 ml of intramuscular inactivated vaccine. (Note: This is also a patient sentence.)
Study Design	A sentence containing information about the design of a study.	A prospective international observational cohort study, with a nested comparative study performed in 349 intensive care units in 23 countries.
Research Goal	A sentence containing information about what a study aims to achieve.	The aim of this study was to investigate the balance between pro- and anti-inflammatory mediators in SA.

useful in presenting more information within the limited screen estate.

3.3 Methodology

We cast the two extractions as a multi-granularity categorization task of two levels, one at sentence-level and the other at word-level:

Key Sentence Classification: We use a five-class scheme as listed in Table 3.4. The first three classes map to PICO elements: *patient* → P, *intervention* → I/C, and *result* → O. In addition, we also have a fourth class, *study design*, which indicates the strength of evidence of a study for users, and a fifth class, *research goal*, which helps them determine whether a study is likely to provide useful information to the clinical questions they have in mind.

Keyword Classification: We use six classes for words as listed in Table 3.5. The first four cover the SCORAP of patient demographics (as described in Table 3.1): *sex* → S, *condition* → CO/P, *race* → R and *age* → A. The last two are introduced to extract the names of intervention and study design.

Table 3.5: Classes for words.

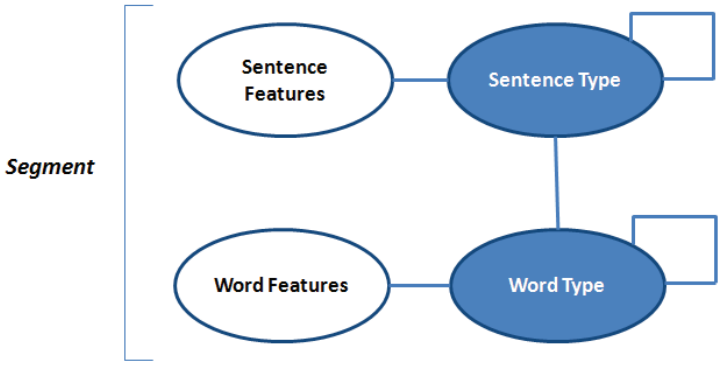
Name	Definition	Example
Sex	The sex of the patients.	male, female
Age	The age (group) of the patients.	54-year-old, children
Race	The race of the patients.	Chinese, Indian, Caucasian
Condition	The condition of the patients, usually a disease name.	COPD, asthma
Intervention	The name of the procedure applied to the patients.	intramuscular inactivated vaccine
Study Design	The name of the design of the study.	cohort study, RCT

These two classifications can be described by the nodes and edges in the segment layer of our correlation graph by instantiating the segment to be sentences and the sub-segment to be words in the sentences as shown in Figure 3.3. If we only consider the correlations at their respective levels, the types of sentences and words can be determined by their features (*i.e.*, observable characteristics) and the types of their neighbours.

This translates into our baseline model, the *independent model*, as shown at the top left corner of Figure 3.4. In this model, the two classifications are performed independently of each other. The words in the same sentence are categorized together and the type of a word is determined based on its features and the types of the other words in the same sentence. In contrast, the sentences are categorized individually based on their own features. We have decided not to consider the types of neighbouring sentences because of two reasons. First, taking the types of neighbouring sentences into consideration would require much more sentences to be annotated and used for training. This would significantly increase the time and effort needed. Second, it would also greatly increase the complexity of one of the models we are going to introduce later. Therefore, the correlation between the type of a sentence and the ones of its neighbours is omitted from all our models.

Nevertheless, as discussed earlier, a suitable technique should address both levels of classifications since they are equally important for the extraction of key information. For this purpose, the correlation between these two classifications, as represented by the edge between sentence type and word type in Figure 3.3, needs to be exploited. These correlations can be observed through a closer

Figure 3.3: Correlations exploited for Resource Categorization on nominal facets.



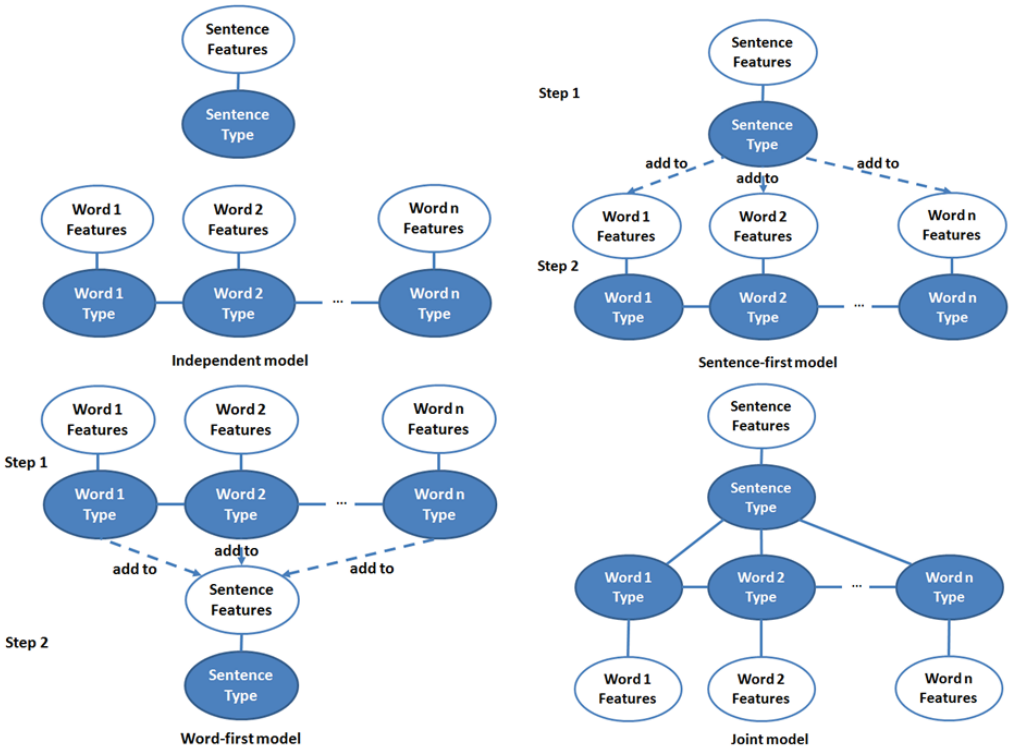
inspection of our classes:

Take the patient class from key sentence classification and the sex, age, race and condition classes from keyword classification as an example. These two sets of classification tasks are correlated: If a sentence is classified as a patient sentence, its words are more likely to represent the patients' sex, age, race and condition. Likewise, if the words in a sentence have been categorized into one of the sex, age, race, condition classes, this sentence is likely to be a patient sentence. Similar correlations can be identified between the study design/intervention sentence class and the corresponding keyword class.

A straightforward approach for exploiting this correlation is to perform the classifications in sequence so that the results from the earlier classification can be incorporated into the later one. This gives rise to the two pipelined models we propose, as shown at the top right and bottom left corners of Figure 3.4. In the *sentence-first model*, key sentence classification is performed first and the resulting sentence class labels are added as evidence for keyword classification (added to the feature vectors of keyword classification as additional features). In the *word-first model*, this process is done in the opposite direction; *i.e.*, keyword classification is performed first and the resulting word labels are added to the feature vectors of key sentence classification.

While these two models are able to incorporate information from the earlier classification into the later one, there is no way for the earlier classification to benefit from the later. Consequently, classification performance can improve on one level but not both. To overcome this problem, we investigate a fourth, *joint*

Figure 3.4: Four models for multi-granularity Resource Categorization of two levels.



model as shown at the bottom right corner of Figure 3.4. It is basically the unrolled version of Figure 3.3 without the looping edge at the sentence type node. In this model, the two levels of classifications are mutually informed of each others' results via joint inference. Therefore, the sentence labels now may influence the prediction of the word labels and vice versa. While often advantageous to performance, the joint model significantly increases the model complexity of the classifier and the training time. Note that if the looping edge were to be included in this model, the resulting model would have been prohibitively expensive to train since it would have contained all the sentences and words from the same article.

In addition, we observe from our inspection that the sentence classes are not mutually exclusive. As shown in Figure 3.2 and the intervention sentence example in Table 3.4, a sentence may contain more than one type of information.

We compare two common approaches to achieve this soft classification. The first is to use train a multi-class classifier on super classes which are the supersets of the existing classes, and classify the sentences into one of these super classes.

Table 3.6: Features for key sentence classification.

Group	Examples
Token	n-grams (sequences of n words, where $1 \leq n \leq 3$) of the sentence.
Sentence	Length of the sentence and its position in the paragraph and in the article.
Named Entity	Whether the sentence contains person name, location name and organization name.
MeSH	Whether the sentence contains MeSH terms and their categories among the 16 top categories of the MeSH tree.
Lexica	Whether the sentence contain a word which appears in the age/sex/race wordlist. All these wordlists contain common words found in the corpus which indicate age, sex and race, respectively.

The classes a sentence belongs to are then all the classes that make up this particular super class. For example, if we only consider the patient and intervention classes, a multi-class classifier can be trained on three super classes: patient, intervention, and patient & intervention. The sentences that are classified into these super classes will be considered to belong to the patient class, the intervention class and both the patient and intervention classes, respectively. The second method is to train one one-against-all classifier for each class: A sentence belongs to a class as long as the corresponding classifier reports positive.

Factoring these two possible approaches into our models, we have eight candidate models in total.

We implement these models using Conditional Random Fields (CRFs), not only because it is the state-of-the-art model for information extraction, but also because its structure can be arbitrarily defined such that different instances from different classification problems can be learned in the same model. This feature allows us to build the necessary joint model. For accuracy concerns, we use an exact inference algorithm: the junction tree algorithm, from the GRMM package[2] for the joint model. We use the MALLET package[3] for the others.

The feature sets for key sentence classification and keyword classification can be found in Table 3.6 and 3.7 respectively. Both feature sets consist of generic text classification features, such as n-grams and named entity information, as well as domain-specific features, such as MeSH terms and class-specific lexica.

As shown in our previous work [Zhao et al., 2010], all the listed features

[2] http://mallet.cs.umass.edu/grmm/
[3] http://mallet.cs.umass.edu/

Table 3.7: Features for keyword classification.

Group	Examples
Token	The word itself, its stem and its part-of-speech tag.
Phrase	Position of the word in the phrase and the head noun of the phrase if it is a noun phrase.
Named Entity	Whether the word is part of a person name, location name or organization name in the sentence.
MeSH	Whether the word is part of a MeSH term and the categories of that term among the 16 top categories of the MeSH tree.
Lexica	Whether the word appears in the age/race/sex wordlist. The wordlists are the ones used in key sentence classification.

contribute positively to the two classifications. For example, token features are crucial to key sentence classification as removing them can lead to significant drop in performance, while MeSH and lexica features play important roles in keyword classification by covering the vocabulary for the classes.

3.4 Evaluation

As part of our effort in developing a domain-specific search system for healthcare (to be detailed in Chapter 6), we have collected 19,893 medical abstracts and full text articles from 17 quality journal websites as recommended by the healthcare practitioners from the Evidence-based Nursing Unit at the National University Hospital. From this collection, 2,000 randomly selected sentences were annotated for the evaluation of key sentence classification.

Within the resulting dataset, there are 220 (11%) sentences in the patient class, 174 (8.8%) in intervention, 448 (22.5%) in result, 119 (6%) in study design, 71 (3.6%) in research goal and 1,329 (66.4%) other sentences not belonging to any of the classes.

For the evaluation of keyword classification, 12,339 tokens (including words and punctuation) from 360 sentences that belong to the patient, intervention and study design classes were annotated. There are 72 (0.6%) words in the sex class, 177 (0.9%) in age, 19 (0.2%) in race, 531 (4.3%) in condition, 607 (4.9%) in intervention, 284 (2.3%) in study design and 10,651 (86.3%) other tokens not belonging to any of the classes.

Considering the fact that the joint model is based on joint inference, which

is computationally expensive in general, we decide to first train the models only on the sentences containing at least one type of key information to obtain a preliminary sense on how the models perform. This is referred to as the *reduced dataset*. Later, we evaluate the models on the *full dataset*, in which a large amount of irrelevant sentences are present as noise, and examine the negative impact of such noise on the classification performance. Lastly, we explore two options: data filtering and feature selection, to alleviate this negative impact of irrelevant sentences and present the final results.

We evaluate the performance of classifiers using the standard information retrieval measures of precision, recall and F_1-measure. 5-fold cross validation is applied in all experiments to avoid overfitting.

3.4.1 Results and Discussions I: Reduced Dataset

The evaluation results for key information extraction using the eight candidate models are listed in Table 3.8, for the reduced dataset. This dataset represents an artificial case where we know *a priori* that a sentence does contain key information and the key sentence classification stage is only used to determine which of the five classes it belongs to.

The general classification performance, as shown in the results of the independent models, indicates that the extractions are precise ($P > 0.8$) for most sentence classes and some word classes. However, there is still much room for improvement on recall for most classes. For key sentence classification, the high precision suggests that a small portion of the sentences from each class can be easily recognized, perhaps because they are written in a conventionalized style. In contrast, the low recall signals that the majority of the sentences – especially those in the intervention, study design and research goal classes – is still hard to detect, possibly due to the variety of linguistic expressions, and the fact that crucial information which determines the sentence's class may be short (1-2 words) in comparison to the length of whole sentence (sometimes on the order of 50 or more words). In our opinion, this performance is acceptable in the context of domain-specific IR as the limited screen estate in the search systems only allows us to show a few sample sentences. For keyword classification, the problem of

Table 3.8: Evaluation results of key information extraction on the reduced dataset using independent (I), sentence-first (SF), word-first (WF) and joint (J) models. (M) and (O) indicate whether the model is based on one multi-class classifier or multiple one-against-all classifiers. The numbers in **Bold** indicate the best P/R/F for a particular class among all the models.

Class\Model	I(M)			I(O)			SF(M)			SF(O)		
	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	.81	.64	.71	.81	.75	**.78**	Same as I(M)			Same as I(O)		
Intervention	.74	.38	.50	**.82**	.47	.60						
Result	.83	**.96**	.89	.90	.95	.92						
Study Design	.93	.42	.58	**.97**	.59	.73						
Research Goal	.89	.47	.61	**.95**	.58	.72						
Keyword Classification												
Sex	.91	**.94**	**.93**	.89	.92	.90	**.91**	.83	.87	.90	.86	.88
Condition	.46	.33	.39	.45	.31	.36	.56	.36	.44	**.60**	.41	**.49**
Race	.82	**.47**	.60	.80	.42	.55	.90	**.47**	**.62**	.82	**.47**	.60
Age	.73	.55	**.63**	.71	**.57**	**.63**	**.82**	.43	.56	.73	.49	.59
Intervention	.57	.33	.42	.59	.34	.43	.65	.28	.39	.77	.35	.48
Study Design	.84	.73	.78	.85	.73	.78	**.94**	.47	.63	.90	.62	.74

Class\Model	WF(M)			WF(O)			J(M)			J(O)		
	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	**.86**	.62	.72	.84	.72	**.78**	.75	.63	.69	.64	**.90**	.75
Intervention	.81	.43	.56	.73	.55	**.63**	.34	.45	.46	.62	**.59**	.61
Result	.82	**.96**	.89	.89	**.96**	**.93**	.83	.94	.88	**.91**	.91	.91
Study Design	.96	.45	.61	.93	.70	**.79**	.65	.72	.54	.83	**.76**	**.79**
Research Goal	.87	.45	.59	**.95**	.58	.72	.32	.46	.62	.86	**.67**	**.76**
Keyword Classification												
Sex	Same as I(M)			Same as I(O)			.88	.69	.78	.88	.71	.79
Condition							.43	**.47**	.45	.59	.36	.45
Race							0	0	0	**1**	.11	.19
Age							**.79**	.43	.56	.76	**.45**	.57
Intervention							.33	.35	.34	.57	.39	**.47**
Study Design							.70	.71	.71	.91	**.75**	**.82**

Table 3.9: Demographics of sentence classes in the multi-class models. P, I, Re, SD and RG stand for patient, intervention, result, study design and research goal respectively.

Single Classes (5)		Duple Classes (8)		Triple Classes (7)		Quadruple Classes (2)	
P	54	P/I	13	P/I/Re	23	P/I/Re/SD	1
I	16	P/Re	50	P/I/SD	9	P/I/SD/RG	13
Re	288	P/RG	13	P/I/RG	6		
SD	18	P/SD	23	P/Re/SD	9		
RG	23	I/Re	64	P/SD/RG	6		
		I/RG	2	I/Re/SD	7		
		I/SD	18	I/SD/RG	3		
		SD/RG	12				
Total	399	Total	195	Total	63	Total	14

linguistic variation also plagues recall for some classes. For example, "children", "45-year-old" and "35 to 40 years of age" are all valid ways of expressing age information. In addition, when the vocabulary size of a class is too large to be effectively covered by medical dictionaries (*e.g.*, condition and intervention), the classification performance is also greatly compromised.

In terms of the relative performance between the multi-class and one-against-all models, the results of the independent models show that the former has a small advantage over the latter in keyword classification (+0.03 to +0.05 on F_1-measure for sex, condition and race) but the latter is better in key sentence classification (+0.03 to +0.15 on F_1-measure for all classes). The inferior performance of the multi-class models on key sentence classification has lead to inferior performance on keyword classification in the sentence-first and joint models, while their advantages on keyword classification do not help them outperform their one-against-all counterparts on key sentence classification in the word-first model or the joint model.

To get a better idea of why the multi-class models do not perform well in key sentence classification, we have carried out a post-hoc analysis on our corpus which reveals the following demographics of the sentence classes in them as shown in Table 3.9.

In total, there are 22 (5 single + 8 duple + 7 triple + 2 quadruple) sentence classes in the multi-class models, 17 (8 + 7 + 2) of which are multiple classes (*i.e.*, consisting of more than one single class). Among these multiple classes, 8 (47%)

of them have less than 10 sentences, 7 (41%) have 10 to 30 sentences, while only 2 (12%) have more than 30 sentences. Moreover, as a result of putting the multi-class sentences into their own multiple classes, 3 of the single classes, namely intervention, study design and research goal now have less than 30 sentences. In contrast, all the one-against-all models have more than 50 sentences as positive examples for the binary classifier of each class. In other words, there are much more classes but much fewer examples in each class in the multi-class models than in the one-against-all models due to the existence of multiple classes in the former. Considering the fact that the multi-class models actually perform slightly better than the one-against-all models on keyword classification where there is no multiple class, we believe the data sparsity caused by the multiple classes is the main reason why the multi-class models perform worse than the one-against-all models on key sentence classification.

While it is possible that the multi-class models may outperform the one-against-all models with a larger corpus, based on our current experiments, the one-against-all models do provide a natural way of handling soft-classification without running into the data-sparsity problem, reduce the computational cost by allowing the classifiers to be trained independently and in parallel, and have shown promising results. Therefore, we believe such models are likely to be practical solutions for Resource Categorization on nominal facets in domain-specific IR and will focus on them only from this point onwards.

Lastly, when it comes to the relative performance of different ways to exploit the correlation between the two categorizations, the sentence-first model outperforms the independent model on the challenging keyword classes such as condition and intervention. However, it also harms the extraction of some of the other keyword classes. Based on our error analysis, we have discovered that when key sentence classification misclassifies a sentence as not containing any key information, it misleads keyword classification into thinking that none of the words in that sentence represent key information. This happens often due to the low recall of key sentence classification. Similarly, in the word-first model, we have also observed that when keyword classification fails to identify the keywords which represent any key information, it misleads key sentence classification into

Table 3.10: Time (in seconds) required for training the independent (I), sentence-first (SF), word-first (WF) and joint (J) models on different percentages of the reduced dataset (671 sentences). All the models are implemented using one-against-all classifiers.

Percentage	I	SF	WF	J
1	0.21	0.17	0.16	15.80
5	0.50	0.50	0.49	184.45
10	1.31	1.33	1.29	369.95
20	3.18	3.20	3.15	794.67
40	8.19	8.11	8.05	1484.13
60	13.63	13.65	13.41	2639.48
80	20.16	20.90	20.00	3039.65
100	28.66	28.54	28.13	3970.60

thinking that the sentence does not contain any key information. Nevertheless, the results from keyword classification are still useful to key sentence classification. This can be seen from the results that the word-first model does improve key sentence classification on most classes in spite of error propagation.

As for the joint model, it is comparable to the rest of the models when the correlation between the sentence and the words is simple. For example, it performs well for the two study design classes, largely because the study design sentences are only concerned with study design words and vice versa. In comparison, it is less effective for the patient sentence class and the four related word classes since the correlations among these five classes are more complex. Nevertheless, it is the only model that can enhance key sentence classification and keyword classification simultaneously. This nature eliminates the need to decide the sequence of the classification tasks and thereby hindering the classification accuracy of the earlier task. However, despite all these advantages, its computational cost is still a major drawback. While the other models can be trained within a minute, the joint model requires up to about an hour.

To get a better understanding of the computational cost of the joint model, we have measured the time required for training it using different percentages of the reduced dataset. The results are as shown in Table 3.10. For comparison purposes, the results for the other three models are also listed.

As can be observed from the table, the joint model does require much more time to train than the other models; however, it scales linearly with the number

of sentences used for training just like the rest. Therefore, we consider it to be expensive but not prohibitively so. With suitable optimization at the implementation level to lower the training cost per sentence, we believe it is still a viable option for practical use.

3.4.2 Results and Discussions II: Full Dataset

As informative as the results on the reduced dataset are, they do not represent the complete picture since both classifications need to be done on all sentences, not just the ones that contain key information. Classification on the full dataset constitutes a real-world trial for both classifiers, as the entire articles are provided. Table 3.11 shows the performance of the models when the full dataset is substituted for the reduced dataset.

The 1,329 sentences added can be considered as noise since none of them contain key information. The presence of such noise adds to the challenge for both classifications and leads to lowered performance for all models in general. Among all the results, only the precision for the intervention, study design and research goal sentence classes are maintained, indicating that the some of the sentences in these classes are still easily distinguishable even in the full dataset.

In the word-first model, the keyword classification results now negatively impact key sentence classification. This is due to the many occurrences of keywords outside of key sentences. The most adversely affected sentence class is the patient class, because it is related to most (four) word classes, all of which can no longer be reliably classified. In contrast, key sentence classification now also functions as a filter for the sentences that do not contain any information instead of just distinguishing the sentences of one class from others. With this classification acting as a filter, it is less likely for the words in the newly added sentences to be misclassified as representing key information. As a result, the sentence-first model returns higher keyword classification performance than in the reduced case. The joint model also suffers a drop in performance. In addition, the training process now requires about 2.5 hours[4], while the rest of the models can be trained within minutes.

[4]Still linear to the number of sentences in the training set.

Table 3.11: Evaluation results of key information extraction on the full dataset using independent (I), sentence-first (SF), word-first (WF) and joint (J) models. All the models are implemented using one-against-all classifiers. The numbers in **Bold** indicate the best P/R/F for a particular class among all the models. The numbers in the brackets indicate the relative performance when compared to the evaluation with reduced dataset (Table 3.8).

Class\Model	I			SF			WF			J		
	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	**.75** (-.06)	.52 (-.23)	**.61** (-.17)	Same as I			.67 (-.17)	.37 (-.35)	.48 (-.30)	.52 (-.12)	**.71** (-.19)	.60 (-.15)
Intervention	**.82** (0)	.34 (-.13)	.48 (-.12)				.58 (-.15)	.38 (-.17)	.46 (-.17)	.58 (-.04)	**.50** (-.09)	**.54** (-.07)
Result	**.78** (-.12)	**.63** (-.32)	**.70** (-.22)				**.78** (-.11)	.60 (-.36)	.68 (-.25)	.77 (-.14)	.58 (-.33)	.66 (-.25)
Study Design	**.97** (0)	.51 (-.08)	.67 (-.06)				.91 (-.02)	.65 (-.05)	.76 (-.03)	.84 (+.01)	**.71** (-.05)	**.78** (-.01)
Research Goal	**.97** (+.02)	.45 (-.13)	.62 (-.10)				**.97** (+.02)	.42 (-.16)	.59 (-.13)	.79 (-.07)	**.63** (-.04)	**.70** (-.06)
Keyword Classification												
Sex	.63 (-.26)	.63 (-.29)	.63 (-.27)	**.74** (-.16)	**.76** (-.10)	**.76** (-.12)	Same as I			.68 (-.20)	.60 (-.11)	.64 (-.15)
Condition	.20 (-.25)	.11 (-.20)	.14 (-.22)	**.53** (-.07)	**.34** (-.07)	**.42** (-.07)				.49 (-.10)	**.34** (-.02)	.40 (-.05)
Race	.62 (-.18)	**.42** (0)	**.50** (-.05)	.83 (+.01)	.26 (-.21)	.40 (-.20)				**1** (0)	.05 (-.06)	.10 (-.09)
Age	.56 (-.15)	**.44** (-.13)	.49 (-.14)	**.66** (-.07)	.42 (-.07)	**.52** (-.07)				.62 (-.14)	.36 (-.09)	.46 (-.11)
Intervention	.46 (-.13)	.25 (-.09)	.32 (-.11)	**.74** (-.03)	.26 (-.09)	.39 (-.09)				.49 (-.08)	**.36** (-.03)	**.42** (-.05)
Study Design	.81 (-.04)	.64 (-.09)	.71 (-.07)	**.93** (+.03)	.59 (-.03)	.72 (-.02)				.86 (-.05)	**.71** (-.04)	**.78** (-.04)

3.4.3 Results and Discussions III: Full Dataset with Data Filtering and Feature Selection

To reduce the noise due to the additional irrelevant sentences and lower the training cost for the joint model, we have also investigated two additional directions: performing data filtering as a preprocessing step to the models and applying feature selection techniques in the training process.

Data Filtering

The idea of data filtering is to remove the negative examples while retaining the positive ones so that both the skewness of data and the data size can be reduced [Gliozzo et al., 2005]. In information extraction (*e.g.*, [Roth and Yih, 2001; Sitter and Daelemans, 2003]), this is commonly done by using a binary classifier to determine whether a segment of text (*e.g.*, a sentence) is likely to contain information of interest. If so, the segment will be processed further for extraction; otherwise, it is filtered.

In our case, we build an additional classifier to filter out the sentences that do not contain any key information. When an unseen sentence is given, this filtering classifier is first applied to determine whether the sentence is unlikely to contain any key information. If so, this sentence and the words in it will be considered as not belonging to any of the sentence or word classes; otherwise the sentence and the words in it will be further classified into the sentence and word classes. With this filtering step, we use only the sentences that belong to at least one of the sentence classes as training data (similar to the reduced dataset case). In this way, the level of noise is minimized and the cost of training the joint model is alleviated.

For consistency, we implement the filtering classifier as a binary classifier using the feature set for key sentence classification. Sentences not belonging to any sentence classes are considered as positive examples, and the rest negative. As shown in Table 3.12, the resulting classifier is able to filter out noise reasonably well (recall for noise > 0.8) but it also incorrectly removes a portion of the key sentences (recall for key sentence < 0.8). Nevertheless, as we are going to show next, the benefit of applying data filtering is already evident with this filtering

Table 3.12: Performance of the filtering classifier.

	P	R	F
Noise	.88	.87	.87
Key sentence	.72	.75	.74

performance. Therefore, we apply this classifier as it is without optimization.

The results after applying data filtering, as shown in Table 3.13, are generally favorable. Improvements can be observed in both key sentence and keyword classifications for most classes. Moreover, the improved keyword classification is able to benefit key sentence classification once again, as indicated by the performance of the word-first and joint models. Last but not least, with data filtering, the joint model only needs to be trained on the reduced dataset. Therefore, the computational resources required for this model remain manageable and unaffected by the size of the full dataset.

Although the resulting performance is still not as good as ones from the reduced dataset, data filtering is easy to implement and able to meet both of our goals. Therefore, we consider it a good choice for key information extraction.

Feature Selection

Ideally, by selecting a good subset of relevant features, both the noise from irrelevant features and the dimensionality of the feature space are reduced. Therefore, the resulting models will be more accurate and take less resources to train. As such, we apply several common feature selection techniques onto the independent model with different percentages of features to retain. The best combination of technique and percentage is then applied to all models to assess its effect.

We have implemented three metrics for computing the importance of the features as is done in [Yang and Pedersen, 1997]:

Document frequency is the number of training instances in which a feature occurs. Features with low document frequency are considered to be non-informative and can be removed.

Mutual information measures the dependence between a feature and a class and is computed using the following formula: $\log{(A \times N)/((A + C) \times (A + B))}$[5].

[5]A: the number of times a feature occurs in an instance from the positive class, B: the

Table 3.13: Evaluation results of key information extraction on the full dataset using independent (I), sentence-first (SF), word-first (WF) and joint (J) models with data filtering. The numbers in **Bold** indicate the best P/R/F for a particular class among all the models. The numbers in the brackets indicate the relative performance when compared to the evaluation with full dataset without data filtering (Table 3.11).

Class\Model	I			SF			WF			J		
	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	.64 (-.11)	.64 (+.12)	**.64** (+.03)	Same as I			**.67** (0)	.60 (+.23)	.63 (+.15)	.49 (-.03)	**.76** (+.05)	.60 (0)
Intervention	**.70** (-.12)	.41 (+.07)	.51 (+.03)				.63 (+.05)	.48 (+.10)	.54 (+.08)	.53 (-.05)	**.54** (+.04)	**.55** (+.01)
Result	.68 (-.10)	**.69** (+.06)	.68 (-.02)				.66 (-.12)	**.69** (+.09)	.68 (0)	**.72** (-.05)	.65 (+.07)	**.69** (+.03)
Study Design	**.92** (-.05)	.55 (+.04)	.68 (+.01)				.90 (-.01)	.68 (+.03)	**.78** (+.02)	.74 (-.10)	**.71** (0)	.73 (-.05)
Research Goal	**.94** (-.03)	.47 (+.02)	.62 (0)				**.94** (-.03)	.47 (+.05)	.62 (+.03)	.93 (+.14)	**.56** (-.07)	**.70** (0)
Keyword Classification												
Sex	.67 (+.04)	.68 (+.05)	.68 (+.05)	**.68** (-.06)	**.69** (-.07)	**.69** (-.07)	Same as I			**.68** (0)	.61 (+.01)	.64 (0)
Condition	.38 (+.18)	.26 (+.15)	.31 (+.17)	.46 (-.07)	**.35** (+.01)	.40 (-.02)				**.52** (+.03)	.33 (-.01)	**.41** (+.01)
Race	.72 (+.10)	.41 (-.01)	.53 (+.03)	.89 (+.06)	**.42** (+.16)	**.57** (+.17)				**1** (0)	.11 (+.06)	.19 (+.09)
Age	.60 (+.04)	.53 (+.09)	.57 (+.08)	**.68** (+.02)	.49 (+.07)	.57 (+.05)				**.68** (+.06)	**.54** (+.18)	**.60** (+.14)
Intervention	.49 (+.03)	.32 (+.07)	.39 (+.07)	**.65** (-.09)	.33 (+.07)	**.44** (+.05)				.51 (+.02)	**.39** (+.03)	**.44** (+.02)
Study Design	.76 (-.05)	.71 (+.07)	.73 (+.02)	**.89** (-.04)	.62 (+.03)	.73 (+.01)				.85 (-.01)	**.72** (+.01)	**.78** (0)

The main caveat of this measure is that it favors the features with lower document frequency and hence is less reliable when the document frequency of the features differs greatly.

Chi-square measures the dependence between a feature and a class by comparing the correlation between them to the χ^2 distribution with one degree of freedom. The formula for this measure is as follows: $N \times (AD - CB)/((A + C) \times (B + D) \times (A + B) \times (C + D))$[6]. This measure is a normalized value and hence is comparable across all features. However, it is less reliable for features with low document frequency because the comparison to the χ^2 distribution would no longer be accurate in that case.

The computed metrics are used to select the top 25%, 50%, 75% of the features for both classifications.

The performance of the independent model after feature selection using different techniques and selection percentages can be found in Table 3.14. (The performance without feature selection is also listed for the ease of reference.)

The effects of feature selection techniques on key sentence classification are mixed. On one hand, feature selection alleviates the problem of low recall we have encountered earlier. The improvement in recall is substantial as less features are selected (*e.g.*, > 0.09 improvement on average with selection by document frequency at 25%). On the other hand, however, this improvement is at the cost of precision, which steadily deteriorates in the selection process. Consequently, the resulting F_1-measure is better than original but only slightly. As mentioned previously, in the context of domain-specific IR, precision is more important than recall. Therefore, feature selection on key sentence classification is not very necessary for our purpose but it can still serve as a way to improve recall in other tasks where a more balanced classification performance is preferred.

In terms of how the feature selection techniques perform with respective to each other, mutual information turns out to be the weakest while document

number of times a feature occurs in an instance from the negative class, C: the number of times a feature does not occur in an instance from the positive class, N: the number of training instances.

[6] D: the number of times a feature does not occur in an instance from the negative class. The rest are the same as the previous footnote.

Table 3.14: Effects of feature selection techniques: document frequency (DF), mutual information (MI), and chi-square (CHI), with different selection percentages on the independent model. The numbers in **Bold** indicate the best P/R/F for a particular class among all technique-percentage combinations.

Class\Selection	DF-25%			DF-50%			DF-75%			No Selection		
Method	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	.62	**.61**	.61	.69	.58	**.63**	.72	.55	.62	**.75**	.52	.61
Intervention	.68	**.42**	.52	.75	.41	**.53**	.80	.36	.50	.82	.34	.48
Result	.73	**.69**	.71	**.78**	.68	**.72**	**.78**	.65	.71	**.78**	.63	.70
Study Design	.84	.63	.72	.90	.58	.70	.94	.56	.70	**.97**	.51	.67
Research Goal	.83	.56	.67	.89	.56	.69	.92	.49	.64	**.97**	.45	.62
Keyword Classification												
Sex	**.63**	.60	.61	**.63**	.58	.60	**.63**	.61	.62	**.63**	.63	.63
Condition	.12	.06	.08	.13	.06	.08	.18	.09	.12	.20	**.11**	.14
Race	.54	.32	.40	.50	.32	.39	**.67**	**.53**	**.59**	.62	.42	.50
Age	.59	.36	.44	**.61**	**.48**	**.55**	.56	.43	.48	.56	.44	.49
Intervention	.46	.20	.28	.43	**.26**	**.32**	.42	.23	.30	.46	.25	**.32**
Study Design	.75	.62	.68	.80	.62	.70	.81	.62	.70	.81	.64	.71

Class\Selection	MI-25%			MI-50%			MI-75%			No Selection		
Method	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	.63	.42	.50	.71	.36	.48	.71	.36	.48	**.75**	.52	.61
Intervention	.76	.38	.50	**.84**	.34	.48	.84	.29	.43	.82	.34	.48
Result	.69	.62	.65	.75	.59	.66	.76	.57	.65	**.78**	.63	.70
Study Design	.80	.71	.75	.88	.65	.74	.90	.60	.72	**.97**	.51	.67
Research Goal	.94	.48	.64	.97	.44	.60	1	.39	.57	**.97**	.45	.62
Keyword Classification												
Sex	.57	**.82**	.67	.57	.82	.67	.58	.82	**.68**	**.63**	.63	.63
Condition	**.25**	.10	**.15**	.20	.07	.11	.22	.09	.12	.20	.11	.14
Race	.54	.37	.44	.54	.37	.44	.54	.37	.44	.62	.42	.50
Age	.56	.35	.43	.52	.28	.36	.54	.28	.37	.56	.44	.49
Intervention	.51	.22	.31	.51	.21	.29	**.53**	.20	.29	.46	.25	**.32**
Study Design	.80	.63	.71	.78	.65	.71	**.82**	**.67**	**.74**	.81	.64	.71

Class\Selection	CHI-25%			CHI-50%			CHI-75%			No Selection		
Method	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	.62	**.61**	.62	.67	.58	.62	.70	.56	.61	**.75**	.52	.61
Intervention	.62	.41	.50	.73	.39	.51	.78	.37	.50	.82	.34	.48
Result	.74	**.69**	**.72**	.77	.67	**.72**	.77	.65	.70	**.78**	.63	.70
Study Design	.77	**.74**	**.76**	.83	.69	.75	.87	.66	.75	**.97**	.51	.67
Research Goal	.72	**.69**	**.71**	.81	.59	.68	.85	.55	.67	**.97**	.45	.62
Keyword Classification												
Sex	.59	.75	.66	.63	.72	.67	.62	.67	.64	**.63**	.63	.63
Condition	.19	.08	.11	.23	.10	.14	.22	**.11**	.14	.20	**.11**	.14
Race	.56	.48	.51	.60	.47	.53	.63	**.53**	.57	.62	.42	.50
Age	.53	.36	.43	.52	.35	.42	.55	.35	.42	.56	.44	.49
Intervention	.49	.22	.30	.50	.21	.29	.49	.23	**.32**	.46	.25	**.32**
Study Design	.78	.60	.68	.78	.59	.67	.77	.64	.70	.81	.64	.71

frequency and chi-square have similar performance. This agrees well with the findings from [Yang and Pedersen, 1997]. Moreover, since document frequency is task-free (*i.e.*, does not require any information about the number of classes and the number of instances in each class), we consider it a suitable choice if feature selection is to be applied on key sentence classification.

In contrast, applying feature selection in conjunction with keyword classification has a negative effect on the performance on all evaluation metrics. This may be due to the fact that the number of features for each word is already small (∼7 on average) and hence applying feature selection would result in too few features for the classifiers to work well.

Since feature selection is not effective on keyword classification, we choose to apply the domain frequency technique with a selection percentage of 25% on key sentence classification to illustrate how it affects the overall performance and efficiency of the four models. The results are shown in Table 3.15.

Compared with the results in Table 3.11, it can be observed that the results for keyword classification in the sentence-first and joint models are not as good as the ones without feature selection. In other words, despite the improvement in recall and overall performance in key sentence classification, it is still more beneficial to keyword classification if key sentence classification is more precise. In this way, keyword classification can rely on key sentence classification to know what types of key information are present in the sentence.

In addition, since feature selection is not done on keyword classification, keyword classification still has an negative impact on key sentence classification as can be seen from the results in the word-first and joint models, which are no better than the independent model with feature selection or the word-first and joint models without feature selection.

Last but not least, the joint model does benefit from feature selection in terms of efficiency. With a low selection percentage, it can be trained with the same level of time as it would require with the reduced dataset.

To sum up, although feature selection may be used to improve recall on key sentence classification and allow the joint model to be trained with less computational resources, it is not applicable to keyword classification and the

Table 3.15: Evaluation results of key information extraction on the full dataset using independent (I), sentence-first (SF), word-first (WF) and joint (J) models with feature selection (the top 25% by document frequency) on key sentence classification. The numbers in **Bold** indicate the best P/R/F for a particular class among all the models. The numbers in the brackets indicate the relative performance when compared to the evaluation with full dataset without feature selection (Table 3.11).

Class\Model	I			SF			WF			J		
	P	R	F	P	R	F	P	R	F	P	R	F
Key Sentence Classification												
Patient	**.62** (-.13)	.61 (+.09)	**.61** (0)	Same as I			.55 (-.12)	.41 (+.04)	.47 (-.01)	.54 (+.02)	**.70** (-.01)	**.61** (+.01)
Intervention	**.68** (-.14)	**.42** (+.08)	**.52** (+.04)				.50 (-.08)	.39 (+.01)	.44 (-.02)	.59 (+.01)	**.42** (-.08)	.49 (-.05)
Result	.73 (-.05)	**.69** (+.06)	**.71** (+.01)				.73 (-.05)	.65 (+.05)	.69 (+.01)	**.74** (-.03)	.58 (0)	.65 (-.01)
Study Design	**.84** (-.13)	.63 (+.12)	.72 (+.05)				.82 (-.09)	**.71** (+.06)	**.76** (0)	.75 (-.09)	.65 (-.06)	.70 (-.08)
Research Goal	.83 (-.14)	.56 (+.11)	.67 (+.05)				**.84** (-.13)	.54 (+.12)	.65 (+.06)	.77 (-.02)	**.65** (+.02)	**.70** (0)
Keyword Classification												
Sex	.63 (0)	.63 (0)	.63 (0)	.66 (-.08)	**.79** (+.03)	**.72** (-.04)	Same as I			**.68** (0)	.69 (+.09)	.67 (+.03)
Condition	.20 (0)	.11 (0)	.14 (0)	.44 (-.09)	**.36** (+.02)	.40 (-.02)				**.53** (+04)	.35 (+.01)	**.42** (+.02)
Race	.62 (0)	**.42** (0)	**.50** (0)	.83 (0)	.26 (0)	.40 (0)				**1** (0)	.11 (+.06)	.19 (+.09)
Age	.56 (0)	.44 (0)	.49 (0)	.61 (-.05)	**.50** (+.08)	**.56** (+.04)				**.68** (+.06)	.37 (+.01)	.48 (+.02)
Intervention	.46 (0)	.25 (0)	.32 (0)	**.59** (-.15)	**.30** (+.04)	**.40** (+.01)				.44 (-.05)	**.30** (-.06)	.35 (-.07)
Study Design	.81 (0)	.64 (0)	.72 (+.01)	**.85** (-.08)	.65 (+.06)	**.74** (+.02)				.81 (-.05)	**.69** (-.02)	**.74** (-.04)

resulting performance is not as good as data filtering. Therefore, we still prefer data filtering over feature selection for our purpose of noise reduction.

3.5 Future Work

Propagating results from one classification to another may do more harm than good if the former cannot be done reliably. This is often referred to as cascading error. As shown in our evaluation, both classifications may mislead each other especially when their accuracy is not good enough or has been compromised due to noise. Ideally speaking, joint inference is a natural way to address this problem because the merging of two classifications into one effectively eliminates the need for propagating results. Nevertheless, in the case where joint inference is not a practical option, a threshold can be set using parameter optimization techniques (*e.g.*, grid search) and only those results whose confidence levels are higher than the threshold are propagated from the earlier classification. This would help to lower the chance of propagating errors into the later classification. From a more fundamental point of view, it is important to improve individual classifications before combining them. To this end, more sophisticated statistical models (*e.g.*, topic models [Steyvers and Griffiths, 2007]) can be investigated in future to manage the endless possible variations of words and sentence structures.

In addition, although the joint model provides a natural way to propagate information between classifications, it incurs a much higher training cost per sentence than other models. Besides reducing the training data size by filtering out irrelevant sentences, we plan to look for more efficient implementations of joint inference algorithms and explore approximate inference algorithms to see if the training cost can be lowered to a manageable level.

This part of our research is done for our domain-specific search system in healthcare (to be detailed in Chapter 6). The extraction and display of key information is the first step in the integration of extraction results into the search process for this system. In future, we also plan to incorporate more sophisticated designs, such as ranking the articles based on how well the query matches with the extracted sentences instead of the whole articles, and filtering the articles

based on the extracted keywords.

3.6 Discussion

The values of the nominal facets of domain-specific resources are usually categories which are distinct from each other. As indicated in our correlation graph, correlations exist between such facets at multiple granularities. In this chapter, we use the problem of key information extraction for evidence-based practice in healthcare as a case study in exploiting such correlations.

In key information extraction, key sentences and keywords need to be extracted from research articles to facilitate applicability and validity assessment. We cast these two extraction tasks as two classification steps and exploit their correlation using models which differ in terms of how information is propagated between them. Our results show that when the two classifications are performed in series, the later classification does benefit from the earlier one. With the help of joint inference, it is possible to propagate information in both directions, such that both classifications simultaneously benefit from each other.

We believe our approach is not limited to healthcare or the specific problem of dual categorizations at sentence-level and word-level. In almost any domain, many other pairs of categories at different granularities are correlated in a similar way and can be tackled likewise. For example, resource type and segment type are correlated because knowing that a domain-specific resource is an encyclopedia page increases the likelihood that one of its segments contains definitions, while knowing that most of the segments in a resource contain research paper information increases the likelihood that it is a journal/conference webpage. We can easily formulate the classifications accordingly and propagate information between them using the proposed models.

Nevertheless, if the granularities of the two categorizations of interest differ significantly (*e.g.*, resource-level vs. word-level), it may be less useful to propagate information between them since the correlation will be much weaker than those whose difference in granularities is smaller (*e.g.*, it would be difficult to determine the type of a resource based on the types of its words and vice versa).

In this case, adding an intermediate level may help to reduce the granularity differences between categorizations and the correlations between them may become strong enough to be exploited (*e.g.*, a segment-level categorization can be added between resource-level and word-level categorizations).

Aside from studying how the correlation between categorizations at different granularities can be exploited, we have also noted and examined two issues related to Resource Categorization on nominal facets.

The first issue concerns soft-classification, where instances may belong to multiple classes. We have compared two approaches: The first is to use one single multi-class classifier for all possible combinations of classes, and the second to use one one-against-all classifier for each class. As observed in our experiments, the former may run into data sparseness problem when there are many possible combinations of classes and when few instances are available for most classes. In contrast, the latter is less affected by this problem since all instances in each class can be used to train the corresponding classifier. In addition, the former only needs to train one classifier while the latter needs to train one for each class and then merge the results of all the classifiers. Therefore, the latter is more difficult to implement but allows the training process to be parallelized. Therefore, to choose between these two approaches, as a way to handle soft-classification in Resource Categorization on nominal facets, we find it important to take note of 1) whether there are sufficient instances in each combination of classes, and 2) whether the training process is costly. When data sparsity is not a concern or a single multi-class classifier can be trained efficiently, the multi-class classifier approach can be taken to simplify the implementation. Otherwise, the one-against-all classifier approach can be used to lessen the adverse effect of data sparsity or parallelize the training process.

The second issue concerns categorization noise. It is common for irrelevant instances to outnumber relevant ones as the granularity of categorization increases. For example, in our corpus, the ratio between sentences that contain key information and those that do not is around 1:2, while the ratio at word-level is 1:6. The presence of noise not only compromises the categorization performance but also increases the computational resources required for training. We

have compared two possible approaches to reduce the noise level in categorization: data filtering and feature selection. Based on our results, we find that both approaches are able to meet the goal of reducing noise and the computational resources required for training; however, data filtering is easier to implement and shows more favorable results, whereas feature selection is able to trade precision for recall with different selection thresholds but is less applicable when the number of features in the instances is already small. Therefore, we believe data filtering is a suitable choice for noise reduction in general. Nevertheless, in the case where the number of features in the instances is large and recall is more important, feature selection can be applied with appropriate selection thresholds for the desired recall level.

Chapter 4

Resource Categorization on Ordinal Facets – A Case Study in Readability Measurement

In this chapter, we examine another class of facets for Resource Categorization – the ordinal facets. These facets establish an ordering of resources based on whether a particular characteristic holds stronger in one than another. Common examples of ordinal facets include: readability [DuBay, 1990], cohesion [Mcnamara et al., 2002] and quality [Wetzler et al., 2009]. Unlike nominal facets, the values of ordinal facets merely indicate the rank of a resource among others.

The ordering established by ordinal facets is particularly useful in domain-specific search systems for sorting the resources so that the ones which have higher values in the desired characteristics can be presented to users first. For example, in medical domain, laymen may be interested in viewing more readable results first, as many medical resources are too specialized for the general public [Graber et al., 1999]. Similarly, uninformed information seekers often prefer resources that are deemed more trustworthy to avoid getting inaccurate or unreliable information. While it is still possible to establish the ordering by performing categorization with an ordered set of labels, any approaches that are able to establish the ordering, such as heuristic-based measurement and ranking, can serve as a viable (and perhaps more natural) way to handle these facets.

Traditionally, ordinal facets are measured using heuristic formula. For example, the Flesch-Kincaid Reading Ease (FKRE) formula [Flesch, 1948] and the Dale-Chall readability formula [Dale and Chall, 1948] are the two most well-

known formula for readability. As another example, the cohesion of a document can be computed by various cohesion metrics (*i.e.*, causal, intentional, temporal and spatial) as is done in Coh-metrix [Mcnamara et al., 2002]. With the development of supervised learning, statistical models can be built based on an annotated corpus and used to compute the values for these facets. For instance, [Collins-Thompson and Callan, 2004] perform a 12-way classification for readability using language models, while [Burstein et al., 2004] assign quality scores to essays on a 6-point scale through linear regression. In the context of domain-specific IR, research efforts have also been made to handle the domain-specific concepts in the resources with the help of domain knowledge sources (*e.g.*, to derive document cohesion based on the amount of semantic relations among the concepts for the computation of readability [Yan et al., 2006]). Despite the improvement in measurement accuracy, the cost of an annotated corpus and domain knowledge sources in those approaches limits their applicability.

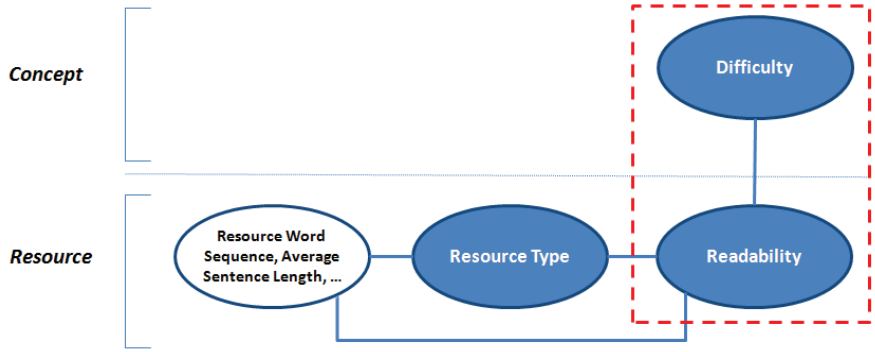
Therefore, in our research, we aim to discover a less expensive (and hence more domain-independent) way of handling domain-specific concepts for the measurement of ordinal facets.

To make our research concrete, we present a case study on readability measurement. There are several reasons why we choose to examine readability.

First, not all ordinal facets are domain-dependent. For instance, *trustworthiness* is often measured for domain-specific resources but the *prestige* (*i.e.*, the level of respects received) of their sources plays a much more important role in the measurement process than domain knowledge. In comparison, readability is more domain-dependent since domain-specific concepts have a strong influence on how readable the resources are. Therefore, investigating this facet may yield more insights on how to handle domain-specificness in a generic manner. Second, while readability measurement for general materials has been well-researched, readability for domain-specific resources has not. This gives us a solid foundation for our work while giving us ample opportunity to improve. Last but not least, readability is a major barrier hindering laymen's understanding of the content in domain-specific resources. As such, an accurate measure of readability is an important component for domain-specific accessibility.

In our correlation graph (Figure 4.1), readability is represented as a node at the resource layer, while its correlations with resource-level observable characteristics, resource type and concept difficulty, are represented as edges to corresponding nodes. Given the fact that domain-specific concepts play a important role in domain-specific readability measurement, our research examines the correlation between resource readability and concept difficulty. Intuitively, this correlation means that resources written for more difficult concepts are less readable while the concepts commonly described by less readable resources are more difficult. We exploit this correlation using an iterative computation algorithm instead of supervised classification to cater for the ordinal nature of both resource readability and concept difficulty.

Figure 4.1: Correlation graph fragment showing nodes and edges relevant to readability. The edge (*i.e.*, correlation) bounded by the dashed line box is examined in this chapter.



We believe our approach is also applicable to other pairs of ordinal facets, such as the specificity of the resources and the genericity of the concepts. We will elaborate on this towards the end of the chapter.

The rest of the chapter is organized as follows. We first review the relevant literature on readability research in Section 4.1. We then describe the intuition behind our approach and how it is embodied in an iterative computation algorithm in Section 4.2. We evaluate our algorithms in the domains of math and medicine in Section 4.3 and point out a few possible directions for future research in Section 4.4. Lastly, we relate our algorithm to several graph-based iterative computation algorithms in Section 4.5 and end with a discussion on Resource Categorization on ordinal facets based on our findings in Section 4.6.

4.1 Literature Review on Readability Measurement

Readability measures indicate how difficult it is to understand a piece of text. Therefore, they are commonly used by educators to select appropriate materials for the target audience. Although they have been applied in many different domains such as education [Flesch, 1948], military [Smith and Senter, 1967] and healthcare [Lay and Florio, 1996], they are mostly generic, *i.e.*, without the flexibility to allow themselves to handle the special elements in any domain. Only recently have researchers started working on domain-specific readability measures. In the following, we first review two major classes of generic readability measures which are based on heuristics and supervised learning respectively, and then move on to domain-specific readability measures.

4.1.1 Heuristic Readability Measures

According to a comprehensive review on classic readability studies [DuBay, 1990], heuristic readability measures were first devised in the 1920s to facilitate the selection of textbooks. They are usually expressed as a weighted sum of the values of some features extracted from a piece of text. The features extracted are the ones that correlate well with readability while their weights are computed by linear regression.

Among all the text features, word features are considered as the strongest predictor. As early as 1923, [Lively and Pressey, 1923] have already demonstrated that the media of the index numbers of the words, as taken from [Thorndike, 1921] which ranks words by their frequencies in a sample text collection, correlate well with readability. Since then, word features have always been a staple in heuristic readability measures. For example, [Vogel and Washburne, 1928] use the number of different words and the number of uncommon words, while [Gray and Leary, 1935] employ the number of different, unfamiliar words.

Other features have been considered as well: [Vogel and Washburne, 1928] also examine five other classes of features, including sentence structure, part of speech, paragraph construction (*e.g.*, the number of sentences), general structure (*e.g.*, the number of lines in a book) and physical makeup (*e.g.*, weight and size

of type). However, among these features, only the number of prepositions and the number of simple sentences are found useful. [Gray and Leary, 1935] further expand the exploration of features by examining 64 countable variables in four categories: content, style, format and features of organization. They identify average sentence length, number of pronouns and number of prepositional phrases as useful in addition to word features.

In 1948, two most succinct yet reliable readability measures were devised: the FKRE formula [Flesch, 1948] and the Dale-Chall readability formula [Dale and Chall, 1948]. Both consist of one sentence feature and one word feature. They share average sentence length as the sentence feature but use the average number of syllables per word and the percentage of words out of a predefined list of 3,000 easy words as the word feature respectively.

Most of the later measures only simplify the computation process. For example, the Automated Readability Index (ARI) [Smith and Senter, 1967] and Coleman-Liau Index [Coleman and Liau, 1975] count characters in a word instead of syllables, while the Simple Measure of Gobbledygook (SMOG) [McLaughlin, 1969] uses the number of polysyllables (*i.e.*, words of more than three syllables) as the only feature. Therefore, up to today, the FKRE and Dale-Chall formula still stand as the state-of-the-art heuristic readability measures.

Although heuristic readability measures provide a quick and indicative way to compute readability, they use only a small number of features to summarize the characteristics of a piece of text. This is often an oversimplification, as much information is lost in the process, such as the identity of the individual words and the knowledge encoded in the text.

4.1.2 Supervised Learning Approaches

To perform readability measurement via supervised learning, one needs to annotate a corpus of text documents with a set of values representing different levels of readability as the training data. Once collected, features can be extracted from the training data to build a model that captures the relationship between the features and the values. Then the resulting model can be used to predict the readability value of an unseen document based on its extracted features.

Under this framework, many researchers have re-examined the utility of most text features. Starting from word features, [Collins-Thompson and Callan, 2004] construct one unigram language model for each of the 12 American grade levels based on a corpus of webpages with grade-level annotations. These language models capture the probability of a word occurring in the document of a certain grade level. The readability of a new document is then predicted by finding the language model that most likely generates all the words in it. Their evaluation shows that this approach outperforms the traditional reading measure on webpages. [Leroy et al., 2008] adopt their approach for classifying health information into three levels (basic, intermediate and advanced), achieving a high accuracy of 98%. Further along this line, [Schwarm and Ostendorf, 2005] explore the effect of using higher order n-gram models (up to trigram) on classification performance and show that it helps to minimize error rates.

Besides using higher order n-gram models, [Schwarm and Ostendorf, 2005] also attempt to combine word features with other text features. They first compute the perplexity scores which indicate how well the language model of the document to be classified matches with the ones built from documents for each of 12 grade levels. These perplexity scores are then used as the feature set of a Support Vector Machine (SVM) classifier together with other text features, such as FKRE score and out of vocabulary rate scores, as well as four parse features, such as average parse tree height and average number of noun phrases. Although the set of non-word features considered is not large, this classifier is able to further minimize the error rates compared to the one based on trigrams.

An alternative approach to combine different types of features is to train one classifier for each type and then fuse their predictions. For example, [Heilman et al., 2007] extend [Collins-Thompson and Callan, 2004] by introducing a k-Nearest Neighbour (kNN) classifier on grammatical features such as sentence length and parse tree patterns. The predictions from the kNN classifier are interpolated with the ones from the SVM classifier to produce a final prediction, which is found to perform better than using either one of the classifiers alone.

Most recently, [Pitler and Nenkova, 2008] examine by far the largest set of textual features. Their feature set includes word (unigram language model),

syntactic (identical to the parse features in [Schwarm and Ostendorf, 2005]), lexical cohesion (*e.g.*, average cosine similarity between sentences), entity coherence (*e.g.*, the transition probability of an entity from being the subject in one sentence to the object in the next) and discourse relations (*i.e.*, language model over discourse relations instead of words). Their results show that word features and average sentence length are strong predictors but the strongest ones are discourse features. Moreover, there is also a complex interplay between different types of features. While successful, their study is a proof-of-concept; they acknowledge that automatic extraction for such rich features does not yet exist.

Despite the fact that supervised learning approaches offer better accuracy compared to heuristic measures, there are still a few issues that limit their utility in domain-specific readability. First, all previous work require an annotated corpus as the training data. This is costly to construct for domain-specific resources, since it requires much domain knowledge to define the facet values and perform the annotation accordingly. Second, although language modeling helps to generate useful word features, it is largely ignorant of the domain-specific concepts. It treats domain-specific concepts as a sequence of words without considering their semantics or the relationships among them. Therefore, it would not be as effective for domain-specific readability measurement.

4.1.3 Domain-specific Readability Measures

To reduce the need for a corpus and better handle domain-specific concepts, domain-specific readability measures have focused on identifying the difficulty of such concepts with domain knowledge. Depending on the type of domain knowledge utilized, these measures can be classified into the following two categories.

Wordlist-based Approaches: The wordlist-based approaches derive the difficulty of domain-specific concepts from domain wordlists. For example, in the domain of consumer healthcare, [Kim et al., 2007] use the familiarity scores from the Open Access and Collaborative Consumer Health Vocabulary (OAC-CHV) as the estimated difficulty. A distance score is computed based on how far an unseen document's vocabulary differs from known document samples. This score is combined with two other distance

scores that are based on text length and syntactic features, to compute a the final readability score. This approach correlates well with the heuristic-based measures on most documents, while correctly identifying the difficult documents which heuristic-based measures miss. However, whether the familiarity features work well compared to other features is left unexamined in their study.

[Borst et al., 2008] associate difficulty with rarity. This is in turn estimated by the size of generic English wordlists (12,000 to 264,000) in which a medical term appears. Their hypothesis is that the smaller the wordlist a word appears in, the more common (and thus less difficult) it is. The complexity of the words in a document is summarized by their average complexity and combined with the average sentence length to produce a final score. An accuracy of 92% is achieved when applied to the two case problem of distinguishing documents targeted at non-experts from ones targeted at medical professionals.

Ontology-based Approaches: In contrast to the wordlist-based approach, ontology-based approaches utilize an existing ontology of domain-specific concepts to derive possible indicators for readability. [Yan et al., 2006] introduce two additional components into the Dale-Chall Readability formula for medical documents: document scope and document cohesion. The document scope is based on the scope of the medical terms in the document. The deeper the terms are in the MeSH hierarchy, the smaller in scope (and hence more readable) a document is. On the other hand, the document cohesion measures the relatedness of the medical terms in a document. The more associations the terms have to each other with respect to the ontology, the more cohesive (and hence more readable) a document is. The combined formula is reported to be significantly better correlated with the readability computed by heuristic readability measures.

In short, these measures address two issues of supervised learning approaches: the need for a corpus and ignorance of domain-specific concepts. However, they still require domain knowledge and incur substantial labor cost in constructing

their annotated wordlist or ontology. These resources may not be available for other domains. As a result, the applicability of such methods remains limited.

All these previous works have refined generic readability measures to be sensitive to nuances within a domain by using manually crafted sources of information. Is there a less expensive way to introduce domain-specific readability?

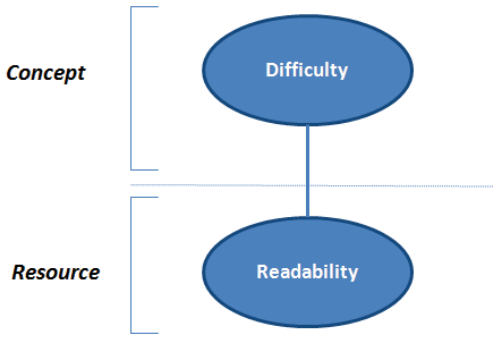
Our method addresses this need. It is based on iterative computation instead of supervised learning and hence does not require definition of facet values or annotation on a corpus. Moreover, it derives concept difficulty from a collection of resources with the help of a concept list, both of which are easily available in any domain. Therefore, our approach outperforms generic readability measures yet remains domain-independent.

4.2 Methodology

Our method exploits the correlation between the readability of domain-specific resources and the difficulty of domain-specific concepts by iterative computation.

In our correlation graph, this correlation is represented by the edge between the difficult node in the concept layer and the readability node in the resource layer as shown in Figure 4.2.

Figure 4.2: Correlation exploited for Resource Categorization on ordinal facets.



This correlation translates into a simple mutually recursive intuition on domain-specific resources and concepts:

- A domain-specific resource A is less readable than another resource B if A is written for more difficult domain-specific concepts than B.

- A domain-specific concept A is more difficult than another concept B if A is described by less readable domain-specific resources than B.

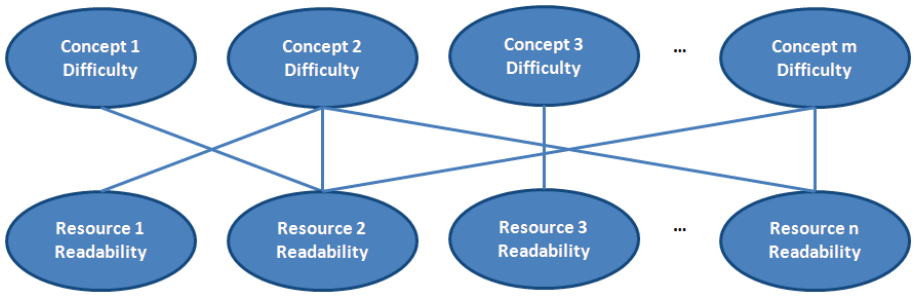
This intuition helps us solve cases where the generic readability measures lead to incorrect conclusions for the difficulty of domain-specific concepts in isolation. For example, let say we need to determine whether a resource written for the concept "ring theory" is less readable than another one written for the concept "Pythagorean theorem". If we extract normal text features such as the average number of syllables or the percentage of familiar words, "Pythagorean theorem" would be incorrectly calculated as more difficult than "ring theory". However, if we examine a corpus of resources written for these two concepts, we may discover that "ring theory" is also described by less readable pages about advanced math concepts, such as "isomorphism theorem" and "abelian group", whereas "Pythagorean theorem" is described by more readable pages about basic math concepts, such as "triangle" and "sine". With this information, we can decide that "ring theory" is more difficult than "Pythagorean theorem" and infer that the resources written for "ring theory" are less readable than ones written for "Pythagorean theorem".

In this way, we can compute the readability of domain-specific resources and the difficulty of domain-specific concepts based on each other.

Unrolling and instantiating the nodes and edges in Figure 4.2, we have a bipartite graph as shown in Figure 4.3 with nodes representing the readability of domain-specific resources and the difficulty of domain-specific concepts respectively. Edges exist between pairs of readability and difficulty nodes if the corresponding resource is written for the corresponding concept. Then we can iteratively compute 1) the value of a readability node based on the values of the adjacent difficulty nodes, and 2) the value of a difficulty node based on the values of the adjacent readability nodes.

Under this iterative computation paradigm, we have experimented with two different ways to compute the values. The first version is heuristic, which models the values as real numbers and computes new values using simple heuristics. Despite the simplicity of this approach, it readily delivers promising results. The second version is probabilistic, which models the values with probability

Figure 4.3: Correlation exploited for Resource Categorization on ordinal facets (unrolled version).



distributions and computes new values using an adapted version of the Naïve Bayes classification. This approach is able to achieve similar performance as the heuristic version with less iterations and increase the expressiveness and flexibility of our algorithm.

The input needed is minimal. Our method requires a list of domain-specific concepts and a corpus of domain-specific resources. A key distinction of our proposal from previous works is that both do not need to be annotated – a flat list of concepts and a corpus of resources are all that is required.

This is an easy requirement to satisfy for most domains: A list of domain-specific concepts is usually available in the form of a domain-specific dictionary, an encyclopedia, or the index at the back of a textbook. Given such a list, a domain-specific corpus can be constructed by downloading the top N (*e.g.*, 100) results of each of the listed concepts from a search engine. Conversely, if a list of domain-specific concepts cannot be found but there are existing collections of domain-specific resources, such collections can be taken directly as the corpus while the list can be constructed by extracting key phrases [Witten et al., 1999] or by simply listing all the noun phrases from it. Lastly, if neither of them exists, one can manually select a small number of domain-specific concepts as a seed list, and then collect a corpus of domain-specific webpages with the help a search engine. One can then iteratively expand them by extracting phrases from the corpus to expand the list and then using the expanded list to collect more webpages for the corpus.

In any case, the amount of domain knowledge needed (*i.e.*, knowing whether a concept belongs to a specific domain) by our approach is significantly less

than the amount needed by other domain-specific readability measures (*i.e.*, to define and assign facet values or to construct a concept ontology). Therefore, we consider our approach to be less dependent on domain knowledge sources and hence more domain-independent.

We describe our approach in detail in the next subsection.

4.2.1 Iterative Computation Algorithm

The first step of our method is to construct a resource-concept graph in the style of the unrolled version of our correlation graph. This graph is bipartite, containing two types of nodes, one representing concepts, the other representing resources. Edges are added between a concept node and a resource node to represent the occurrence of the former in the latter. After constructing this graph, we start score computation by first assigning an initial readability score to each resource node and a difficulty score to each concept node. We can then iteratively update the readability scores for the resources based on the difficulty scores of the associated concepts (and vice versa) until the termination condition is met. The final scores at the resources nodes are their readability values.

The details of the graph construction and the two versions of score computation are as follows:

Graph Construction

Given a list of concepts and a collection of resources, the construction of the graph proceeds as shown in Algorithm 4.1: We create a representing concept node for each concept in the list (Line 2-5) and a representing resource node for each resource in the collection (Line 6-9). We then add an edge between a concept node and a resource node if the concept represented by the former occurs on the resource represented by the latter (Line 10-13). This completes the construction of graph and Figure 4.4 gives an example of a graph constructed based on two resources and a list of concepts.

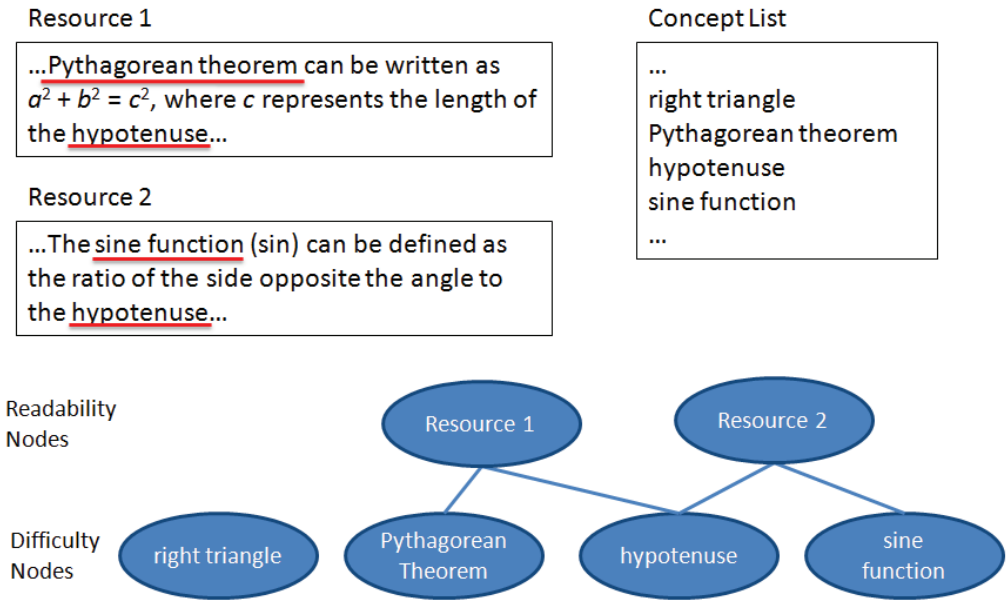


Figure 4.4: Example of graph construction.

Algorithm 4.1 construct-graph(*conceptList*, *corpus*)

```
1: create graph G
2: for each concept c ∈ conceptList do
3:     create concept node cNode
4:     cNode.c = c
5:     add cNode to G.C
6: for each resource r ∈ corpus do
7:     create resource node rNode
8:     rNode.r = r
9:     add rNode to G.R
10: for each resource node rNode ∈ G.C do
11:     for each concept node cNode ∈ G.C do
12:         if occur(cNode.c, rNode.r) then
13:             add edge (rNode,cNode) to G.E
14: return G
```

Score Computation

After graph construction, score computation can begin. Both heuristic and probabilistic versions of this stage follow the same general flow and can be sub-divided into three steps: initialization, iteration and termination. In the initialization step, we assign an initial score to each resource node, representing its readability, and to each concept node, representing its difficulty. Then we move on to the iterative computation step in which the new score of each node is computed. At the end of each iteration, we check whether the termination condition is met. If so, the scores of the nodes will be updated a final time as the new scores and the

computation terminates; otherwise, the update is followed by more iterations until the termination condition is finally met. Upon termination, the scores the resource nodes are the computed readability values.

The details of heuristic and probabilistic score computations are as follows:

Heuristic Score Computation

The pseudocode for the initialization step of heuristic score computation is shown in Algorithm 4.2.

Algorithm 4.2 heuristic-initialize(G)

```
1: for each resource node rNode in G.R do
2:     rNode.score = FKRE(rNode.r)
3: for each concept node cNode in G do
4:     cNode.score = 0
5:     for each resource node rNode in adj(rNode) do
6:         cNode.score+ = rNode.score
7:     cNode.score/ = size(adj(rNode))
8: normalize(G.R)
9: normalize(G.C)
```

We initialize the scores of the resource nodes using the FKRE formula (as shown below) since it is one of the classic, widely-used heuristic readability formula as described in Section 4.1.1.

$$score_{rNode} = FKRE(r) = 206.835 - 1.015 * avgSL_r - 84.6 * avgWL_r \quad (4.1)$$

where $avgSL_r$ and $avgWL_r$ stand for the average sentence length in words and the average word length in syllables of the resource r respectively [Flesch, 1948].

For a concept node, we initialize its score as the average readability of all the resources containing the concept (Line 3-7) as shown in the following equation.

$$score_{cNode} = \frac{\sum_{rNode \in adj_{cNode}} score_{rNode}}{|adj_{cNode}|}, \quad (4.2)$$

where adj_{cNode} stands for the collection of nodes adjacent to $cNode$.

We then proceed to the iterative computation step (Algorithm 4.3), in which the new score of each node is computed as the average of the scores of the

Algorithm 4.3 heuristic-iterate(G)

1: **for each** node n in G **do**
2: $n.newScore = 0$
3: **for each** node $aNode$ in $adj(n)$ **do**
4: $n.newScore += aNode.score$
5: $n.newScore = n.newScore/size(adj(n)) + n.score$
6: normalize($G.R$)
7: normalize($G.C$)

neighboring nodes plus its current score:

$$newScore_n = \frac{\sum_{aNode \in adj_n} score_{aNode}}{|adj_n|} + score_n, \tag{4.3}$$

where $adj(n)$ stands for the collection of nodes adjacent to the node n.

(Note: The scores are normalized after initialization and each of the iterative computation step.)

Lastly, the termination check (Algorithm 4.4) is done by computing the change in the ranks of the resource nodes based on their scores to see if it stabilizes (*i.e.*, smaller than the selected threshold). We take the square root of the residual sum of squares (RSS) divided by the number of nodes as a measure of change. Specifically, this change is computed using the following formula:

$$change = \sqrt{\frac{(\sum_{rNode \in R}(newRank_{rNode} - rank_{rNode})^2)}{|R|}} \tag{4.4}$$

where R stands for the collection of resource nodes in the graph.

Algorithm 4.4 terminate(G)

1: $change = 0$
2: $RSS = 0$
3: **for each** resource node $rNode$ in G **do**
4: $rNode.newRank = rank(rNode.newScore)$
5: $rNode.rank = rank(rNode.score)$
6: $RSS = RSS + (rNode.newRank - rNode.rank)^2$
7: $change = (RSS/size(G.R))^{1/2}$
8: return $(change < THRESHOLD)$

An example of this heuristic score computation (normalization omitted for clarity's sake) can be found in Figure 4.5.

The convergence of heuristic score computation can be established in a way similar to the proof of convergence for the HITS algorithm. For this purpose,

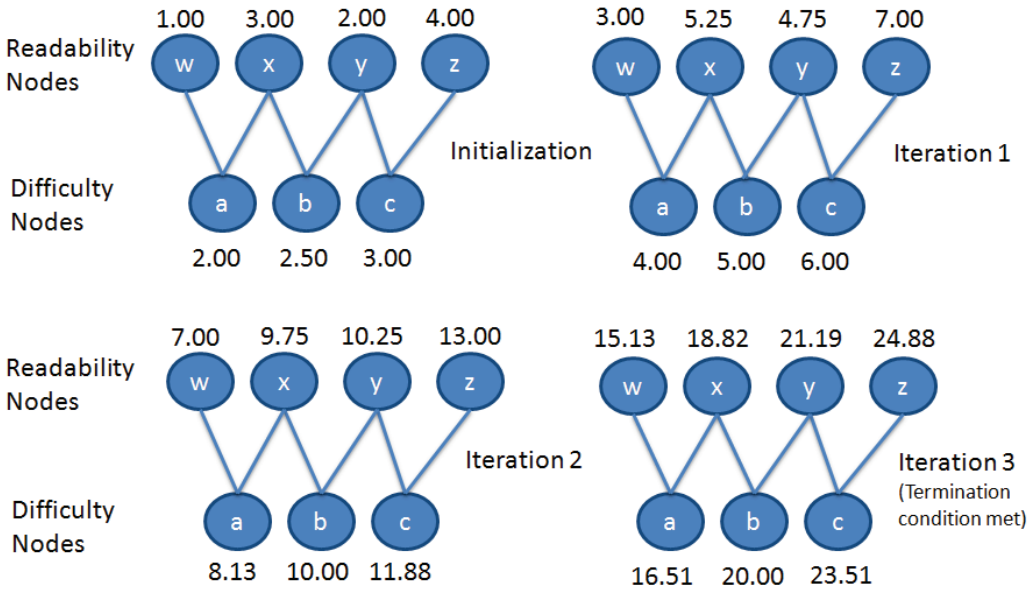


Figure 4.5: Example of heuristic score computation. Normalization is omitted for clarity's sake.

we introduce the following alternative notations from linear algebra to describe the computation process.

First, we index the resources with the integers in $[1...|R|]$ where $|R|$ is the total number of resources. Then we represent the readability scores of all the resources as the column vector $X = \{score(rNode_1), ..., score(rNode_{|R|})\}^T$ where $score(rNode_i)$ is the readability score for the resource node $rNode_i$ corresponding to resource i. Since this vector changes over iterations, we use X_0 to denote the state of this vector after initialization and X_k for the state after iteration k.

Similarly, we index the concepts with the integers in $[1...|C|]$ where $|C|$ is the total number of concepts. Then we represent the difficulty scores of all concepts as the column vector $Y = \{score(cNode_1), ..., score(cNode_{|C|})\}^T$ where $score(cNode_i)$ is the difficulty score for the concept node $cNode_i$ corresponding to concept i. We also use Y_0 to denote the state of this vector after initialization and Y_k for the state after iteration k.

Second, we encode the adjacency information from the perspective of the resources as a $|R| \times |C|$ matrix A_r. The entry a_{ij} in this matrix is: $\frac{1}{|adj(rNode_i)|}$, where $|adj(rNode_i)|$ stands for the number of nodes adjacent to the resource node $rNode_i$ corresponding to resource i, if the concept node corresponding to concept j is adjacent to $rNode_i$, or 0 otherwise.

Similarly, we encode the adjacency information from the perspective of the concepts as a $|C| \times |R|$ matrix A_c. The entry a_{ij} in this matrix is $\frac{1}{|adj(cNode_i)|}$, where $|adj(cNode_i)|$ stands for the number of nodes adjacent to the concept node $cNode_i$ corresponding to concept i, if the resource node corresponding to resource j is adjacent to $cNode_i$, or 0 otherwise.

For example, the adjacency information in Figure 4.5 can be encoded in the following two matrices if the readability nodes correspond to resource 1, 2, 3 and 4 and the difficulty nodes correspond to 1, 2, and 3, both from left to right.

$$A_r = \begin{vmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{vmatrix} \tag{4.5}$$

$$A_c = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{vmatrix} \tag{4.6}$$

With the above notations, the initialization of readability scores and difficulty scores can be expressed as Equation 4.7 and 4.8.

$$X_0 = \{FKRE(r_1), ..., FKRE(r_{|R|})\}^T \tag{4.7}$$

where $FKRE(r_i)$ is the FKRE score for resource i.

$$Y_0 = A_c X_0 \tag{4.8}$$

As shown in Equation 4.7, X_0 is simply a vector containing the FKRE scores of the resources. As for Y_0, it is computed as A_c multiplied by X_0 so that its position i contains the dot product of row i in A_c and X_0. Since row i in A_c contains $\frac{1}{|adj(cNode_i)|}$ at the positions corresponding to the resource nodes which are adjacent to $cNode_i$ and 0 otherwise, the result is the sum of the readability scores of the adjacent resource nodes multiplied by $\frac{1}{|adj(cNode_i)|}$, which is the average readability scores of these nodes as described in Equation 4.2.

Similarly, the iterative computation step can be expressed as follows:

$$X_k = A_r Y_{k-1} + X_{k-1} \tag{4.9}$$

$$Y_k = A_c X_{k-1} + Y_{k-1} \tag{4.10}$$

As can be observed from these two equations, in the iterative computation process, we are left-multiplying X_{k-1} with A_c to get Y_k, which is then left-multiplied with A_r to get X_{k+1}. In other words, we effectively left-multiply X_{k-1} with $A_r A_c$ to get X_{k+1}. When such multiplication is performed repeatedly as k approaches infinity, this is equivalent to the power iteration method [Wikipedia, 2012b] which states that the vector X_k converges to the dominant eigenvector of the matrix $A_r A_c$. Therefore, the readability scores converge.

Probabilistic Score Computation

Although the heuristic score computation algorithm we have just introduced serves as a simple yet effective way to perform the score computation, its main caveat is that the scores are represented by single values. As a result, it lacks the expressiveness required to model the case where the difficulty of a domain-specific concept is a wide spectrum. For example, simple geometry can be taught to primary school students through geometric shapes, while more challenging aspects of geometry, such as differential geometry, are studied by university students and math experts. Single values, as used in heuristic score computation, would not be able to provide an accurate representation of these difficulty spectrums. Furthermore, the use of single values also loses some flexibility in choosing suitable computational mechanisms. Many sophisticated computational mechanisms, such as Bayesian-style treatments, Expectation Maximization and Belief Propagation, fit naturally into the iterative computation paradigm. However, since they are all probabilistic in nature. it is not possible to incorporate them into heuristic score computation.

To overcome these limitations, in probabilistic score computation, we introduce probability distributions to the nodes in the graph. They represent

the probability of the domain-specific resources and concepts having particular readability and difficulty values respectively. Correspondingly, instead of the scores, these distributions are initialized, iteratively updated and used through the computation process. It is only when single-valued scores are required (*e.g.*, counting the number of resources at a particular readability level and ranking in termination check) that they are converted into scores by taking expectations.

All the distributions are non-parametric because we expect the types of distributions to differ greatly depending on the types of the resources and concepts. Under this formulation, we initialize these distributions through sampling (Algorithm 4.5). For the readability distributions of domain-specific resources, we first take M_{R_S} (= 300) random samples of $N_{R_S}\%$ (= 75) of the sentences in the resources. Then we use the FKRE formula (Equation 4.1) to compute the readability values of these samples, which in turn form the readability distributions of the resources. More specifically, the probability of a domain-specific resource at a particular readability level is calculated as follows:

$$\mathbb{P}_r(rd) = \frac{M_{rd,r}}{M_{R_S}} \tag{4.11}$$

where $M_{rd,r}$ is the number of sentence samples taken from resource r with readability rd and M_{R_S} is the total number of samples.

The sampling for the difficulty distributions of domain-specific concepts is similar: we take M_{C_S} (= 300) samples of $N_{C_S}\%$ (= 75) of the resources in which the concepts appear. Then we take the average readability values of the resources in each sample to form the difficulty distributions of the concepts. The corresponding formula is as follows:

$$\mathbb{P}_c(df) = \frac{M_{df,c}}{M_{C_S}} \tag{4.12}$$

where $M_{df,c}$ is the number of resource samples of concept c whose average readability equals to df and M_{C_S} is the total number of samples.

In the process of trying out different values for the number of samples and proportion of resources sampled, we have observed that lowering sample proportion (*e.g.*, 25%) has a negative impact on the performance. In our opinion,

this may be due to the fact that sampling too few sentences in a resource (or too few resources for a concept) is not representative enough for the calculated readability (or difficulty) to be reliable. In contrast, changing the number of samples does not have much effect on the performance. Nevertheless, we have decided not to use too small a value to allow for more varied distributions. This is why we have chosen the values as mentioned above.

In addition, we quantize the calculated readability and difficulty to K ($= 7$) levels and apply add-one smoothing on the resulting distributions to avoid the data sparseness problem. Note that this choice of 7 as the number of levels of quantization is meant to coincide with the number of readability levels in the annotation of the data we have for evaluation. In practice, it can be tuned to a suitable number which is sufficiently large such that the information loss due to quantization is minimal and yet data sparsity is not an issue.

Algorithm 4.5 probabilistic-initialize(G)

```
1:  K = 7
2:  M_{R_S} = 300
3:  for each resource node rNode in G.R do
4:     for x = 1 to M_{R_S} do
5:        sample = sampleSentence(rNode)
6:        level = quantize(FKRE(sample),K)
7:        rNode.distr[level] + +
8:     for level = 1 to K do
9:        rNode.distr[level]/ = M_{R_S}
10:    rNode.distr = smooth–normalize(rNode.distr)
11:    rNode.score = expectation(rNode.distr)
12: M_{C_S} = 300
13: for each concept node cNode in G.C do
14:    for x = 1 to M_{C_S} do
15:       sample = sampleNeighbour(cNode)
16:       level = quantize(averageFKRE(sample),K)
17:       cNode.distr[level] + +
18:    for level = 1 to K do
19:       cNode.distr[level]/ = M_{C_S}
20:    cNode.distr = smooth–normalize(cNode.distr)
21:    cNode.score = expectation(cNode.distr)
```

Afterwards, we employ an adapted version of the Naïve Bayes classification to iteratively update the readability distributions of the resource nodes based on the difficulty distributions of their neighboring concept nodes, and vice versa.

In standard Naïve Bayes classification [Manning et al., 2008], the probability

of a document *doc* being in category *cat* is computed as:

$$\mathbb{P}(cat|doc) = \mathbb{P}(cat) \prod_{1 \leq k \leq n_{doc}} \mathbb{P}(t_k|cat) \tag{4.13}$$

where $\mathbb{P}(cat)$ is the prior of the category *cat* and $\mathbb{P}(t_k|cat)$ is the conditional probability of the term t_k occurring in a document of category *cat*.

Fitting this formula into our context, we can compute the updated probability of a resource r being at readability level rd as:

$$\mathbb{P}'_r(rd) = \mathbb{P}(rd|r) = \mathbb{P}(rd) \prod_{c \in C_r} \mathbb{P}(c|rd) \tag{4.14}$$

where $\mathbb{P}(rd)$ is the prior of readability rd, C_r is the set of concepts occurring in r, and $\mathbb{P}(c|rd)$ is the conditional probability of the concept c occurring in a resource of readability rd.

The computation of readability priors is straightforward:

$$\mathbb{P}(rd) = \frac{M_{rd,R}}{M_R} \tag{4.15}$$

where $M_{rd,R}$ is the number of resources which are at readability level rd and M_R is the total number of resources.

However, since the conditional probability in Equation 4.14 only models the occurrences of concepts, rather than their difficulty values, we replace it with another conditional probability $\mathbb{P}(df_c|rd)$, which denotes the likelihood of a concept c of difficulty df occurring in a resource of readability rd. This conditional probability is calculated using the following formula:

$$\mathbb{P}(df_c|rd) = \frac{M_{R_{rd},C_{df}}}{M_{R_{rd},C}} \tag{4.16}$$

where $M_{R_{rd},C_{df}}$ is the total count of the concepts at difficulty level df occurring in the resources at readability level rd, and $M_{R_{rd},C}$ is the total count of all the concepts occuring in the resources at readability level rd.

With this replacement, the computation for readability distributions becomes dependent on the difficulty levels of the domain-specific concepts, which are in turn determined by their difficulty distributions. The final formula for readability

distribution computation is as follows:

$$\mathbb{P}'_r(rd) = \mathbb{P}(rd|r) = \mathbb{P}(rd) \prod_{c \in C_r} \mathbb{P}(df_c|rd) \tag{4.17}$$

where $\mathbb{P}(rd)$ is the prior of readability rd as defined in Equation 4.15, C_r is the set of concepts occurring in r, and $\mathbb{P}(df_c|rd)$ is the conditional probability of a concept c of difficulty df occurring in a resource of readability rd as defined in Equation 4.16.

The computation of difficulty distributions closely mirrors its counterpart:

We first calculate the difficulty priors and the conditional probability of a resource r of readability rd containing a concept of difficulty df as shown below:

$$\mathbb{P}(df) = \frac{M_{df,C}}{M_C} \tag{4.18}$$

where $M_{df,C}$ is the number of concepts which are at difficulty level df and M_C is the total number of concepts.

$$\mathbb{P}(rd_r|df) = \frac{M_{C_{df},R_{rd}}}{M_{C_{df},R}} \tag{4.19}$$

where $M_{C_{df},R_{rd}}$ is the total count of the resources at readability level rd containing the concepts at difficulty level df, and $M_{C_{df},R}$ is the total count of all the resources containing the concepts at difficulty level df.

Then the updated difficulty distributions are computed as follows:

$$\mathbb{P}'_c(df) = \mathbb{P}(df) \prod_{r \in R_c} \mathbb{P}(rd_r|df) \tag{4.20}$$

where $\mathbb{P}(df)$ is the prior of difficulty df as defined in Equation 4.18, R_c is the set of resources containing c, and $\mathbb{P}(rd_r|df)$ is the conditional probability of a resource r of readability rd containing a concept of difficulty df as defined in Equation 4.19.

Similar to Equation 4.20, we also change the conditional probability to $\mathbb{P}(rd_r|df)$ so that the computation for difficulty distributions becomes dependent on the readability distributions of domain-specific resources. The pseudocode for this iterative computation step can be found in Algorithm 4.6.

As for the termination condition check, we reuse the one from heuristic score computation (Algorithm 4.4) since the change in ranking can still be computed based on the scores of the domain-specific resources.

A proof for the convergence of probabilistic score computation is challenging because Naïve Bayes classification does not lend itself to a closed-form analysis (*e.g.*, cannot be naturally described using linear algebra). Nevertheless, we believe it also converges due to the way the counting is done in the calculations of priors and conditional probabilities. A sketch of our reasoning is as follows. As mentioned earlier, we take the expectations of the distributions whenever single-valued scores are required to be derived from the distributions. For example, in the calculation of readability priors, the readability levels of the resources are counted based on the expectations of their readability distributions (*e.g.*, a resource is counted as of readability level 4 if the expectation of its readability distribution is close to 4). In this way, we influence the computation by injecting the information that these expectations are considered to be "good" estimates of the readability values. As a result, in the updated readability distributions, the probabilities of the resources being at the corresponding readability levels are increased while others decreased. This leads to the expectations of the distributions getting closer to those levels and the resources being counted again towards those levels. As this process repeats, the readability distributions would eventually be updated to a point where they capture such information well and are no longer affected by it. At this point, the readability distributions converge and so do the readability scores of the resources.

We have just described our iterative computation algorithm for readability measurement given a list of concepts and a collection of resources. In the case where new resources and concepts are added after the iterative computation is completed for the existing resource collection and concept list, we can update the graph structure accordingly, initialize the scores of the newly added nodes based on their adjacent nodes, and then carry out further iterative computations on the updated graph until the termination condition is (again) met. Alternatively, we may re-run the algorithm on the enlarged resource collection and concept list. This should provide more accurate estimation especially when the number

of newly added resources and concepts is substantial.

There are already several well-established algorithms in web search for computing quality scores for webpages such as PageRank, HITS, and SALSA. However, as far as we know, our work is the first to apply this methodology for domain-specific readability measurement. We will relate our approach to the existing graph-based iterative computation algorithms in Section 4.5.

4.3 Evaluation

The goal of our evaluation is to demonstrate the efficacy, robustness and domain independence of our approach.

To accomplish this goal, we have performed three sets of experiments in two different domains. We first evaluate our approach with a collection of math resources and concepts to show its efficacy. Second, since a truly robust method should work well without requiring much domain-specific resources and concepts, we have also investigated into how many math resources and concepts our method needs to achieve good performance. Last, we evaluate the performance of our approach on medical documents to show its domain independence. We discuss these evaluations in turn.

4.3.1 Experiments in Math

While our technique is minimally supervised, to properly assess the results, we need to first compile a set of materials that have gold-standard readability annotations. To ensure fairness, we have sought additional annotators for our main math corpus. The resulting construction, annotation and validation of the ground truth have taken three man-months. We feel that this is a significant investment and would be a data bottleneck for other comparative work. As such, to encourage comparative work, we have made the resulting corpus and judgments available for download[1]. Part of this corpus is also used in Chapter 5 to study the problem of Text-to-Construct Linking in math, and in Chapter 6 for the math search system we have built.

[1] http://wing.comp.nus.edu.sg/downloads/mwc

Table 4.1: Math concepts used in corpus collection.

Type	Concepts
Areas (12)	Arithmetic, chaos theory, differential geometry, discrete mathematics, geometry, linear algebra, modular arithmetic, number theory, numerical analysis, non-parametric statistics, set theory, trigonometry.
Operations (3)	Fourier transform, matrix diagonalization, Monte Carlo method.
Theorems (4)	Bayes' theorem, De Morgan's law, Pythagorean theorem, ring isomorphism theorem.
Objects (8)	Absolute value, bipartite graph, complex number, Dirichlet integral, fraction, function, non-stationary time series, polynomial.

Our corpus of math resources is extended from our earlier work [Zhao et al., 2008]. In total, we have chosen 27 common math concepts from MathWorld, covering different types of math concepts, such as areas (*e.g.*, geometry and number theory), operations (*e.g.*, Fourier transform), theorems (*e.g.*, Pythagorean theorem) and objects (*e.g.*, complex number), as listed in Table 4.1. We chose them specifically to reflect the diversity of concepts in math and ensure the webpages collected cover a wide spectrum of readability.

For each chosen math concept, we performed a Google web search[2] and incorporated the math webpages from the first 100 results into our corpus. The resulting corpus contains 2,381 webpages in total. To obtain the ground truth readability judgments for evaluation, we asked 30 undergraduate students to annotate the readability level for 120 randomly chosen, manually segmented webpages from our corpus. Other dimensions of the webpages were also annotated, but the discussion of these dimensions is out of the scope of this thesis, and hence they are not mentioned further. The details of the readability levels used can be found in Table 4.2.

Subjects were first shown an annotation guide explaining how to use our annotation system and what the readability levels are. After reading the guide, the subjects annotated each webpage by reading it and selecting an appropriate readability level for it as shown in Figure 4.6. Each subject was asked to annotate 20 webpages in 45 minutes and was given a token remuneration as as appreciation

[2] On 21st Nov, 2008.

Table 4.2: Readability levels for webpages.

Value	Corresponding Education Background
1	Primary
2	Lower Secondary
3	Higher Secondary
4	Junior College (Basic)
5	Junior College (Advanced)
6	University (Basic)
7	University (Advanced)

for the effort. On average, each webpage was annotated by 5 to 8 subjects. We took the average annotated values to establish the ground truth of readability.

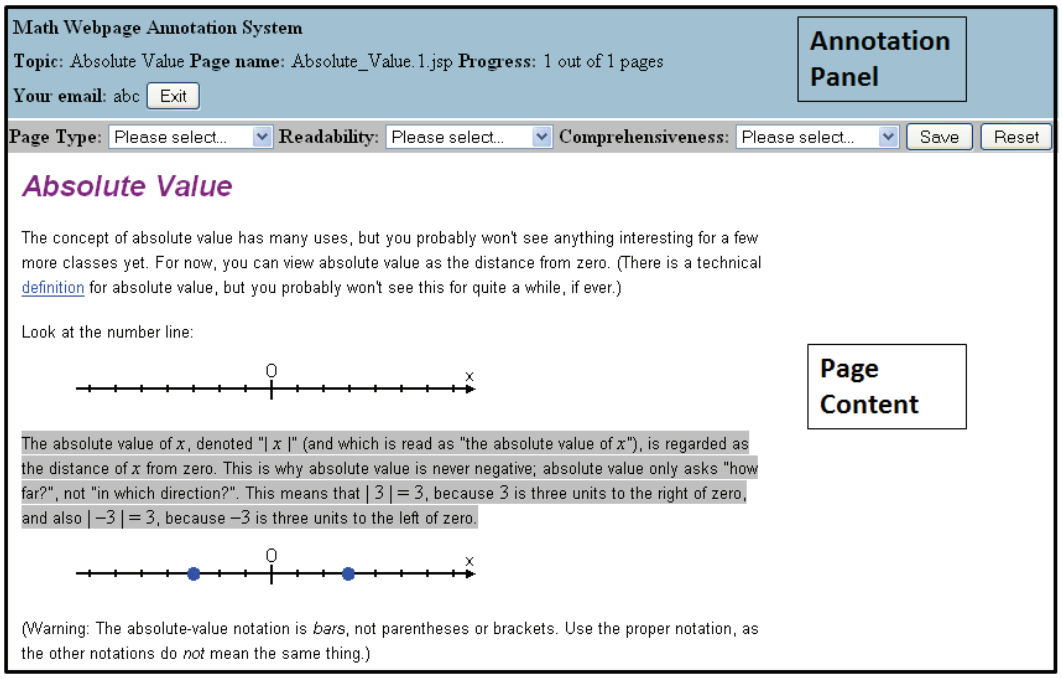


Figure 4.6: Webpage annotation interface: Subjects select a readability level for a webpage from the drop-down menu at the annotation panel.

Before the experiment, we needed to determine whether manual readability annotation is indeed a feasible and reproducible task. To do so, we assessed inter-annotator reliability by computing the pairwise inter-judge agreement using Cohen's Kappa coefficient [Cohen, 1960]. Cohen's Kappa measures the agreement between two annotators, accounting for chance agreement. Its values range from 1.0 (complete correlation/agreement) to -1.0 (complete disagreement/negative correlation). A zero value indicates no correlation. The average pairwise inter-judge agreement is 0.72, indicating substantial agreement. We also applied Fleiss' Kappa [Fleiss, 1971], a multi-rater agreement measure,

to calculate the agreement among all the subjects. The result is similar (0.73).

Since the measured agreement is substantial but not strong (not above 0.80), we manually examined the annotations to discover which levels were being confused. We observed that although the subjects were able to determine what is readable and what not, the exact values annotated might still differ slightly between subjects. This is shown by the fact that 67% of the disagreed readability annotations have a standard deviation of less than 0.5. To eliminate these small perturbations, we applied Spearman's rho [Spearman, 1987], which converts the values to rank order. The measured correlation is 0.93 (again, read on a -1.0 to $+1.0$ scale). This indicates a strong correlation for rank order and confirms our hypothesis that the general order of readability can be reliably distinguished.

After obtaining the gold-standard readability annotations, we proceed to evaluate our approach by pairwise judgement accuracy. For each pair of webpages in the collection, we examine their readability scores from the subjects and those from our system. A pairwise judgement is said to be correct if both scores agree on whether one is more (or less) readable than the other. This metric is chosen instead of precision, recall and F_1-measure because it is more important to be able to determine the relative order between pairs of documents rather than assigning exact labels for ordinal facets like readability.

Not all the pairs of annotated readability values are used for the evaluation. We ignored those whose difference is smaller than a threshold (0.5) – we considered such pairs indistinguishable even by our subjects and hence not suitable to be included into evaluation. In total, there are 5,165 qualified pairwise judgements for the annotated webpages.

Besides pairwise judgement accuracy, we also use Spearman's rho to evaluate how close the overall ranking produced by an approach is to the one established by the ground truth.

General Evaluation

We run our system with the 2,381 webpages in our corpus and a list[3] of 5,861 math concepts compiled from MathWorld. We present the results of our itera-

[3] Also used in Chapter 5 in our experiments on Text-to-Construct Linking.

Table 4.3: Evaluation results on math webpages.

	Pairwise	Spearman
FKRE	.72	.48
NB	.72	.52
SVM	.80	.70
MaxEnt	.82	.67
HIC	**.87**	**.75**
PIC	.85	.73

tive computation algorithm with heuristic and probabilistic score computations (denoted as HIC and PIC respectively) as well as four baselines in Table 4.3.

The four baselines include one standard heuristic measure (FKRE) and three supervised learning approaches: NB[4], SVM[5] and MaxEnt[6]. The classifiers are trained on the annotated webpages using only binary features indicating whether a particular math concept appears on the webpage. We intentionally limit these baseline classifiers to use the same inputs as our system, as we are only interested in how well they could make use of the concepts to perform readability measurement. We also tried adding discretized versions of average word length, average sentence length and FKRE score into the baselines' feature sets, but this did not manage to improve their performance. For all the supervised learning approaches, we perform 5-fold cross validation to avoid overfitting.

As can be seen from Table 4.3, FKRE shows a modest amount of correlation (0.72/0.48) on pairwise judgment accuracy and Spearman's rho respectively). This is similar to the results achieved by NB (0.72/0.52). In contrast, the two other baselines, SVM and MaxEnt, perform much better, scoring 0.80/0.70 and 0.82/0.67 on the two metrics. However, our approach still outperforms all the baselines: 0.87/0.75 for HIC and 0.85/0.73 for PIC.

Furthermore, although the two versions of score computation do not differ much in terms of performance, PIC improves over its heuristic counterpart by reducing the number of iterations required for convergence: The former takes only 7 iterations to terminate on average, while the latter takes 18. A close inspection on how the performance changes as the computation proceeds (Figure 4.7) reveals that the measured pairwise judgment accuracy and Spearman's rho after

[4] http://www.cs.waikato.ac.nz/ml/weka/
[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[6] http://maxent.sourceforge.net/

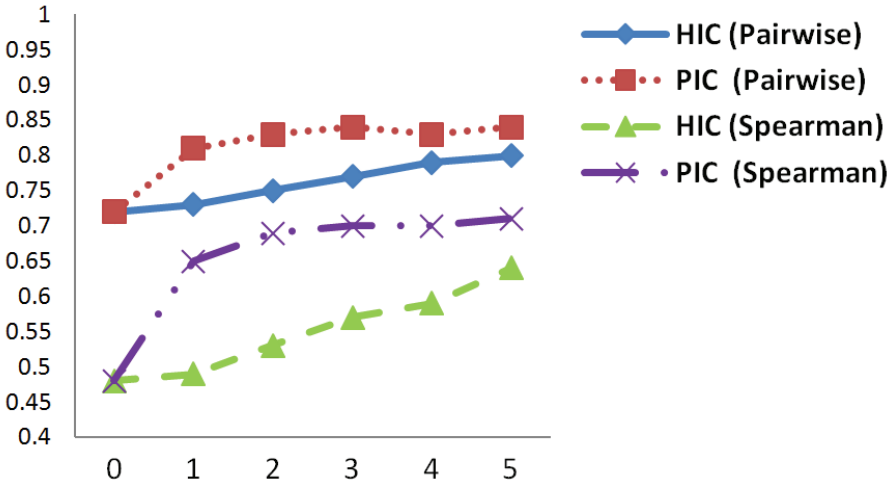


Figure 4.7: Performance of HIC and PIC in the first five iterations.

the initialization step do not differ in both versions; however, PIC achieves much better performance (+.08/+.16) right after the first iteration and quickly moves to convergence. This indicates that the improvement in convergence speed is due to the incorporation of Naïve Bayes classification, a stronger computational mechanism than simple heuristics, into the score computation process. While not directly contributing to the performance, the probabilistic formulation itself, without which the incorporation would not be possible, also deserves some credits on this.

We believe these results strongly validate the efficacy of our method.

Evaluation with Selection Strategies

This second set of experiments is to verify the robustness of our approach. For this purpose, we run our algorithm on subsets of webpages and concepts selected by four different selection strategies: 1) selecting N webpages at random, 2) selecting the top N webpages with the highest quality, as indicated by their ranks in the search results from which they were collected, 3) selecting N concepts at random, and 4) selecting the top N concepts with the greatest importance, as indicated by their total TF.IDF (*i.e.*, the sum of the TF.IDFs of a concept in all webpages). The N mentioned in the selection strategies is set to five different levels: 20%, 40%, 60%, 80% and 100%. The resulting performance of HIC and PIC with these strategies is shown in Figure 4.8 to 4.11. For the experiments

involving random selection, the average performance of five runs is shown.

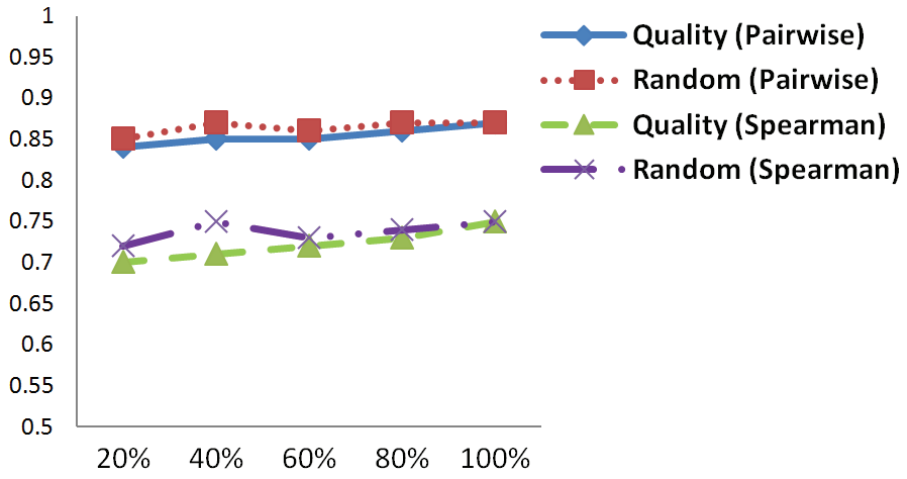


Figure 4.8: Effects of webpage selection strategies on HIC.

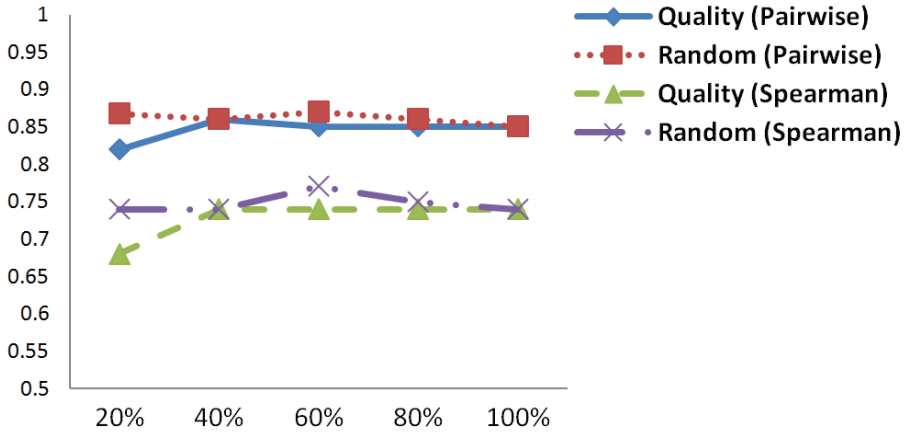


Figure 4.9: Effects of webpage selection strategies on PIC.

Both heuristic and probabilistic score computations exhibit a similar nature when coupled with the selection strategies. Two points are consistent and noteworthy from the results: First, selecting more webpages only improves the performance of our system slightly. Moreover, webpage selection by quality yields no better results than random selection. In other words, our method can work with a small set of webpages without any specific selection strategy[7].

Second, when concepts are selected at random, increasing the number of concepts also helps to improve the performance slightly. However, if the concepts are chosen by importance, using fewer concepts, in fact, further boosts the performance of our system. This indicates that the concepts with low TF.IDF do

[7]The results and conclusions presented here differ from our earlier publication [Zhao and Kan, 2010] due to further optimization of our system.

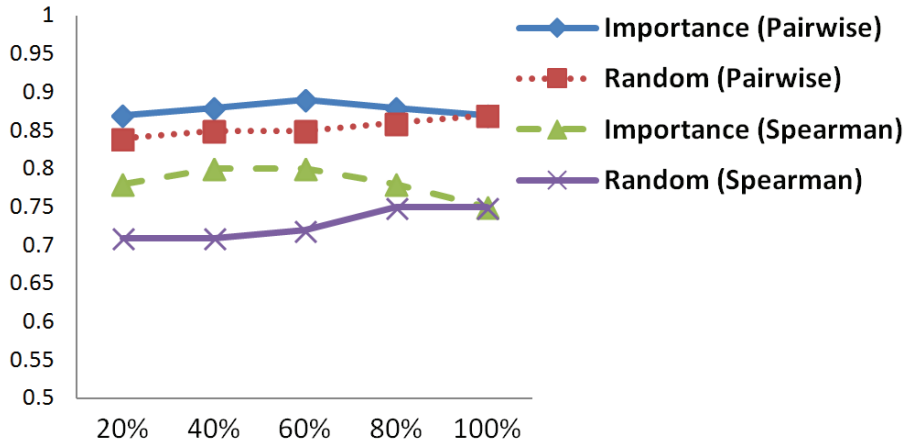


Figure 4.10: Effects of concept selection strategies on HIC.

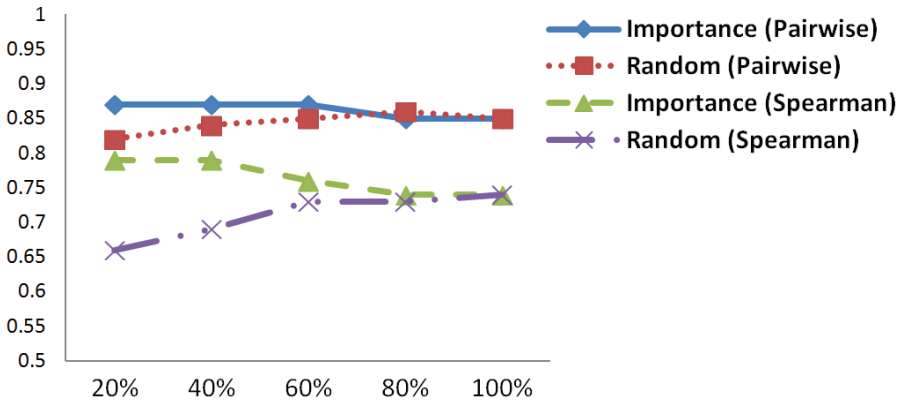


Figure 4.11: Effects of concept selection strategies on PIC.

Table 4.4: Evaluation results on math webpages with selection strategies.

	Pairwise	Spearman
HIC	.87	.75
PIC	.85	.73
HICS	**.89**	**.80**
PICS	.88	.78

not contribute positively to the performance and should be removed from the graph using this selection strategy. Therefore, we have also incorporated the concept selection by TF.IDF into our system and denoted this improved version as HICS (HIC with Selection) and PICS (PIC with Selection). The resulting performance is further improved to 0.89/0.80 for HIC and 0.88/0.78 for PIC, as shown in Table 4.4.

In short, this experiment shows that our approach is robust enough to work with a small set of domain-specific resources and concepts to achieve good performance with simple, automatic selection strategies.

4.3.2 Experiment in Medical Domain

To verify the domain independence of our approach, we repeat our first experiment in the medical domain following the same general methodology. We first selected 27 medical concepts of different difficulty levels and types from MeSH, such as diseases (*e.g.*, diabetes), injuries (*e.g.*, bruise), substances (*e.g.*, vitamin), symptoms (*e.g.*, snoring), therapies (*e.g.*, blood transfusion) and procedures (*e.g.*, bronchoscopy), as listed in Table 4.5. For each of these concepts, we then downloaded the top 100 search results[8] and consolidated the webpages (2,642 in total) for our medical corpus. A subset of the corpus (946 pages) was annotated using the same set of readability levels.

General Evaluation

We run our system with the 2,642 medical webpages in our corpus and a list of 22,792 medical concepts compiled from MeSH. The results are listed in Table 4.6. In this experiment, there are 320,976 pairwise judgments.

[8] On 22nd Oct, 2009.

Table 4.5: Medical concepts used in corpus collection.

Type	Concepts
Diseases (13)	Allergy, cancer, chronic fatigue syndrome, dengue fever, dermatomyositis, diabetes, erysipelas, farsightedness, hepatitis, HIV, leukemia, osteoporosis, thrombocytopenia
Injuries (2)	Bruise, pressure ulcer.
Substances (2)	Aflatoxin, vitamin.
Symptoms (6)	Cough, diarrhea, headache, lordosis, overweight, snoring,
Therapies (2)	Blood transfusion, foraminotomy.
Procedures (2)	Bronchoscopy, magnetic resonance angiography

Table 4.6: Evaluation results on medical webpages.

	Pairwise	Spearman
Heuristic	.63	.28
NB	.73	.53
SVM	**.82**	**.70**
MaxEnt	.76	.60
HIC	.74	.53
PIC	.75	.55
HICS	.75	.53
PICS	.76	.57

The performance of our approach for the medical domain is modest in comparison to the math domain. On one hand, our system still performs much better than the heuristic measure: For HIC, pairwise judgement accuracy and Spearman's rho improve from 0.63/0.28 to 0.74/0.53 (0.75/0.53 after concept selection) respectively. PIC also achieves similar improvement: 0.75/0.55 (0.76/0.57 after concept selection). On the other hand, when compared to the supervised classifiers, our approach performs about the same as NB and MaxEnt but does not manage to outperform SVM. Nevertheless, considering the fact that our approach does not have access to the large amount ($\sim$1000) of readability annotations as the supervised classifiers do, we consider it as performing reasonably well and believe this test does validate its domain independence.

4.4 Future Work

While we are satisfied with our approach, a detailed analysis on the experiment results has revealed several potential areas for improvement:

First, with the adapted version of Naïve Bayes classification as the computational mechanism, the difficulty distributions in probabilistic score computation quickly converge to single difficulty levels. This convergence contradicts one of the motivations for introducing the probabilistic formulation, which is to handle the situation where the difficulty level of a concept varies greatly depending on context. One possible solution to this problem is to split the concept nodes based on the types of context as indicated by the associated resources. In this way, each of the resulting nodes (with the corresponding subset of resources) represents the concept in a particular type of context and their difficulty levels can be readily represented by single values.

Second, as we did not preprocess the webpages to identify their main contents, concepts that are presented as auxiliary information, such as navigation links and advertisements, have added substantial noise to the graph construction process. For example, the math concept "number theory" happens to appear at the navigation panel of MathWorld. Consequently, all the 39 MathWorld pages in our corpus, which make up about 10% of the pages containing the math concept, are included into the difficulty computation for this concept. Similarly, in the medical corpus, there is a webpage about snoring whose main content is written for less than 20 medical concepts. However, it lists more than 100 medical concepts in its navigation bar. As a result, many unrelated medical concepts have been added to the readability computation for this webpage. In both cases, the accuracy of our approach is adversely affected. We believe that further preprocessing to exclude certain sections of the webpages would significantly reduce the number of errors.

Lastly, another factor that compromises our system is the relatively limited spectrum of readability levels in the medical corpus, in comparison to math. Although we have intentionally chosen concepts of different difficulty levels and types, none of the retrieved webpages are targeted at primary school students. This is rather different from the math scenario, where we can easily find highly readable webpages full of games and animations that explain easy math concepts to younger audiences. Without such webpages, our algorithm is limited in its ability to discern and boost basic readability scores. This suggests that the

effectiveness of our algorithm in a particular domain is positively correlated to the width of the readability spectrum.

4.5 Related Graph-based Iterative Computation Algorithms

Our approach is inspired by other successful iterative graph algorithms which have made their impact in digital libraries. We relate and contrast our approach to three of them: PageRank, HITS and SALSA.

PageRank is a link analysis algorithm based on the intuition that the number of backlinks is a good indication of popularity or importance. It works on a graph which contains nodes representing webpages (or publications) and directed edges representing the hyperlinks (or citations) from one node to another. The score of a node is computed as the probability of visiting this node by following the edges randomly. This algorithm has been very successful and widely used in areas such as web search and citation analysis. However, in our problem, there are two types of objects: resources and concepts, with edges representing occurrences. As such, a node with more links means it is a resource that exhibits a larger number of different concepts or a concept that has a higher domain frequency. Due to the fact that the readability of a resource depends on the number of "difficult" concepts instead of the number of different concepts, while the difficulty of a concept tends to be inversely correlated with its domain frequency, we believe a direct application of PageRank would not work for our problem.

HITS is more similar to our algorithm than PageRank in the sense that it also keeps track of two separate hub and authority scores and uses them to compute each other iteratively. The main difference between HITS and our approach is that we consider two types of objects and attach the two readability and difficulty scores separately. In addition, HITS constructs the graph online using a subset of the documents from the corpus retrieved by a query, whereas our algorithm constructs the graph offline with all the documents in the collection.

SALSA combines the strength of PageRank and HITS by incorporating the backlink information into the hub and authority computation. However, the idea

of using backlinks as an indication of readability or difficulty does not have a good parallel in our application.

4.6 Discussion

The ordinal facets of domain-specific resources are different from their nominal counterparts in the sense that their values are meant to establish an ordering. As such, it is harder to define a set of facet values to be assigned to the resources or used as labels for categorization as is done for nominal facets. Moreover, the fact that the constraints placed on these facet values are often relative makes categorization a less natural solution compared to other approaches, such as heuristic-based measurement and ranking.

As a common example of ordinal facets, readability is no exception to these properties. Furthermore, it is also one of the highly domain-specific ordinal facets due to its correlation with domain-specific concepts. Therefore, we choose it as the subject of investigation in our research.

Following our correlation graph, we associate resource readability with concept difficulty and exploit the correlation between them using an iterative computation algorithm. As shown in our evaluation, this algorithm is effective, robust and domain-independent.

Although initially developed for readability, we believe our approach also works for other ordinal facets of domain-specific resources as long as some moderately correlated ordinal facets can be identified for them. For example, we can measure the specificity of a resource by checking the genericity of the concepts it is written for. The intuition behind is that the more generic concepts a resource is written for, the less likely it would be specific enough to cover every aspect of them. On the other hand, the more often a concept is described in many highly specific resources, the less likely this concept is a generic one since most of it can be well explained within a single resource. As another example, we can exploit the correlation between the trustworthiness of domain-specific resources and the prestige of its sponsors. The intuition for this case is that a domain-specific resource is trustworthy if it is from a prestigious source or cites many

prestigious sources, while a source is prestigious if it produces many trustworthy domain-specific resources or is cited by many of them.

In the case where only weakly-correlated ordinal facets and/or correlated nominal facets exist for a targeted facet, our recommendation is to treat it as a nominal facet and apply supervised learning so that all the correlated facets can be taken into consideration. Nevertheless, our method can still be applied to facilitate the computation of the correlated facets to indirectly improve the measurement of the targeted facet. For example, [Wetzler et al., 2009] have shown that "appropriateness for age range" and "has prestigious sponsors" are two of the seven effective indicators for the quality of educational resources. Although not directly applicable to the quality facet itself, our approach can still be employed to estimate the readability of the resources and the prestige of the sponsors to provide information for the computation of the two indicators.

To sum up, we have explored Resource Categorization on nominal and ordinal facets in the previous and this chapter respectively. Due to the extent of this problem, we use two case studies, key information extraction and readability measurement, to illustrate how our correlation graph can guide the categorization process. We have also demonstrated how our approaches can be applied to improve the categorization performance and yet remain domain-independent.

Algorithm 4.6 probabilistic-iterate(G)

```
1: K = 7
2: for rd = 1 to K do
3:    rdPriors[rd] = 0
4: for each resource node rNode in G.R do
5:    rd = expectation(rNode.distr)
6:    rdPriors[rd] + +
7: rdPriors = smooth–normalize(rdPriors)
8: for rd = 1 to K do
9:    for df = 1 to K do
10:      dfGivenRd[rd][df] = 0
11: for each concept node cNode in G.C do
12:   df = expectation(cNode.distr)
13:   for each resource node rNode in adj(cNode) do
14:      rd = expectation(rNode.distr)
15:      dfGivenRd[rd][df] + +
16: for rd = 1 to K do
17:   dfGivenRd[rd] = smooth–normalize(dfGivenRd[rd])
18: for each resource node rNode in G.R do
19:   for rd = 1 to K do
20:      rNode.newDistr[rd] = rdPriors[rd]
21:      for each concept node cNode in adj(rNode) do
22:         df = expectation(cNode.distr)
23:         rNode.newDistr[rd]* = dfGivenRd[rd][df]
24:   rNode.newDistr = smooth–normalize(rNode.newDistr)
25:   rNode.newScore = expectation(rNode.newDistr)
26: for df = 1 to K do
27:   dfPriors[df] = 0
28: for each concept node cNode in G.C do
29:   df = expectation(cNode.distr)
30:   dfPriors[df] + +
31: dfPriors = smooth–normalize(dfPriors)
32: for df = 1 to K do
33:   for rd = 1 to K do
34:      rdGivenDf[df][rd] = 0
35: for each resource node rNode in G.R do
36:   rd = expectation(rNode.distr)
37:   for each concept node cNode in adj(rNode) do
38:      df = expectation(cNode.distr)
39:      rdGivenDf[df][rd] + +
40: for df = 1 to K do
41:   rdGivenDf[df] = smooth–normalize(rdGivenDf[df])
42: for each concept node cNode in G.C do
43:   for df = 1 to K do
44:      cNode.newDistr[df] = dfPriors[df]
45:      for each resource node rNode in adj(cNode) do
46:         rd = expectation(rNode.distr)
47:         cNode.newDistr[df]* = rdGivenDf[df][rd]
48:   cNode.newDistr = smooth–normalize(cNode.newDistr)
49:   cNode.newScore = expectation(cNode.newDistr)
```

Chapter 5

Text-to-Construct Linking

Plain prose text often falls short as a medium of communication in domain-specific resources due to the complexity of the information to be encoded. For instance, the definition of a quadratic equation in text is "an expression of the second degree constructed from variables and constants, using only the operations of addition, subtraction, multiplication, and non-negative integer exponents". This is considerably longer than its math expression counterpart $ax^2 + bx + c$. Similarly, the spatial structure of a chemical compound can be effectively summarized as a structural formula instead of paragraphs of texts. As mentioned in Chapter 2, in our research, we refer to such symbolic representations, which encode domain knowledge through a domain-specific way other than natural language, as *domain-specific constructs*. In practice, they are widely used in domain-specific resources as a more efficient way to convey information.

Despite their superiority in conciseness, domain-specific constructs are hard to deal with in domain-specific IR because they are both structurally and semantically more complex than text. Syntactically, domain-specific constructs take on both textual and graphical forms. For example, in chemistry, chemical formula encode the type and number of the constituent element in a compound through chemical symbols and numeric subscripts (*e.g.*, $C_6H_{12}O_6$), while structural formula represent the spatial arrangement of atoms and bonds by specifying the layout of the atoms and how they are connected to each other. Semantically, the meaning of a symbol may differ greatly depending on the context. For example, variables in math may be used to refer to any math concept – numbers, points, angles and even whole propositions. Without proper annotation, sufficient do-

main knowledge and a suitable internal representation, such information may be lost during indexing. Consequently, there might not be sufficient information to decide whether two constructs match, and thus retrieval performance may suffer.

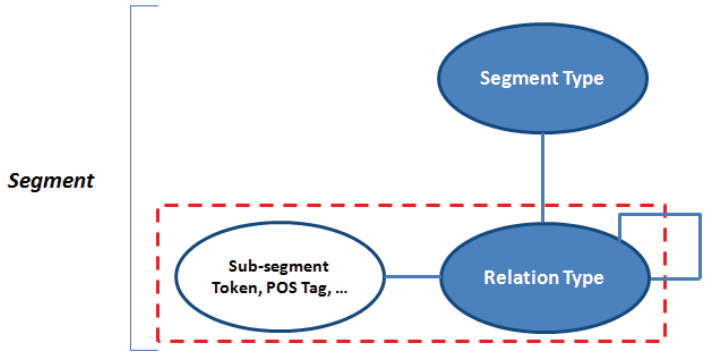
Even if the indexing problem can be solved, domain-specific construct input is still difficult and far less convenient. As mentioned in Section 2.1, more accessible construct input methods, such as plain text and graphical user interfaces, are limited in expressiveness, while more expressive methods, such as specialized markup languages, have steep learning curves. Moreover, based on our user study (detailed in Section 2.2), keyword search is most preferred due to its simplicity and effectiveness.

To make the searching and relevance ranking of constructs relevant to users while maintaining the usability of keyword search, a domain-specific search system needs to be able to perform Text-to-Construct Linking (*i.e.*, to resolve domain-specific concepts to related domain-specific constructs). With this linking ability, the search system may immediately return the linked constructs to users and use them for retrieval.

Text-to-Construct Linking is partly a relation extraction problem in the sense that we can consider a concept and a construct linked if a certain semantic relation, such as the construct being a symbolic representation of the concept, exists between them. Moreover, since one concept may be related to multiple constructs, there is a need to perform construct selection to automatically decide which constructs shall be presented to users and used in retrieval.

In our correlation graph (Figure 5.1), the relation extraction aspect of Text-to-Construct Linking is captured by the nodes and edges in the segment layer. The node of interest for this aspect is the relation type node for the sub-segments. As indicated by the edge between this node and the one representing sub-segment level observable characteristics, the type of relation exists between a concept-construct pair can be determined based on its observable characteristics. Therefore, our proposed approach is to apply supervised learning as a generic way to combine various types of observable characteristics for the prediction of relation type. After extraction, we consolidate the extracted relations and perform

Figure 5.1: Correlation graph fragment showing nodes and edges relevant to relation type. The edge (*i.e.*, correlation) bounded by the dashed line box is examined in this chapter.



construct selection using a simple heuristic function in the style of TF.IDF.

We investigate the problem of Text-to-Construct Linking in the domain of math. In this domain, the texts and constructs to be linked are math concepts, such as absolute value and Pythagorean theorem, and math expressions, which are combinations of numbers, math symbols and operators, such as $\frac{3}{4}$, a and $a^2 + b^2 = c^2$.

The motivation for using math as the target domain is two fold. Math expressions are one of the most common types of domain-specific constructs, and yet an efficient way of handling them has not been discovered so far. In addition, they are frequently written inline and largely text-based. Therefore, conventional relation extraction approaches are likely to be applicable to them and the findings of our research would be more applicable to constructs of similar nature (*e.g.*, DNA sequences and molecular formula).

Although math expressions are commonly inline and textual, we believe our approach can be extended to handle non-inline and/or graphical constructs. We will elaborate on this towards the end of the chapter.

The rest of the chapter is organized as follows. In Section 5.1, we present our literature review on relation extraction and describe the insights we have gained through our corpus study. Then we formulate the problem of Text-to-Construct Linking concretely in Section 5.2. We detail our approach in Section 5.3 and the evaluation results in Section 5.4. Afterwards, we discuss the limitation of our research and propose directions for future research in Section 5.5. We end with

a discussion on Text-to-Construct Linking based on our findings in Section 5.6.

5.1 Background

This section consists of two parts. In the first part, we review existing work on relation extraction. In the second, we detail our corpus study which helps to formulate the problem of Text-to-Construct Linking in the domain of math.

5.1.1 Relation Extraction

Relation extraction is one branch of information extraction that identifies semantic relations between extracted entities. It has been studied extensively and applied on various types of texts, such as plain text [Agichtein and Gravano, 2000], news articles [Doddington et al., 2004], Wikipedia pages [Suchanek et al., 2007] and research articles [Krallinger et al., 2011].

The relations of interest may be binary (*i.e.*, relations between two entities) or multi-way (*i.e.*, relations among more than two entities, *a.k.a*, "events"). Two recent examples are the slot-filling task in TAC '11 [Entity Linking, 2011] which targets 26 binary relations for persons (*e.g.*, country_of_birth and member_of) and 16 for organizations (*e.g.*, members and countries_of_headquarters), and the GENIA event extraction task in BioNLP '11 [Kim et al., 2011] which aims to recognize 9 types of bio-molecular events (*e.g.*, binding and localization) possibly involving multiple proteins/entities at multiple sites.

The approaches for extracting binary relations can be broadly classified into the following two categories:

Rule-based approaches: The rule-based approaches for binary relation extraction are similar to the ones for entity extraction as mentioned in Section 3.2, except that the patterns are defined around two entities and the actions are to report the corresponding relations for the patterns matched. A few examples of these approaches can be found in [Jayram et al., 2006; Shen et al., 2007; Krishnamurthy et al., 2008]. Please refer to Section 3.2.1 for a review on the strengths, weaknesses, and issues of these approaches.

Statistical approaches: In statistical approaches, the extraction of relation

is done by classifying whether the relation of interest exists between a pair of entities using statistical models. There are two groups of methods which differ in terms of how pairs of entities are modeled. The first group of methods models each pair of entities individually as a vector of features. The strength of this approach is that various types of features, such as lexical features, syntactic features and semantic features [Kambhatla, 2004; GuoDong et al., 2005], can be easily cast into a unified framework and employed to comprehensively describe the entities and the context between/surrounding them. As shown in [Jiang and Zhai, 2007], which systematically explore several types of features including entity attributes (*e.g.* entity types), n-grams, constituency-based parse tree features (*e.g.*, grammar productions) and dependency parse tree features (*e.g.*, dependency relations and paths), good performance can be readily achieved using only the basic features from each type. Nevertheless, the fact that many statistical models assume the independence of features and these features can only take on single values, leads to the difficulty in capturing structured information, such as parse trees.

The second group of methods defines similarity between pairs of entities using a kernel function. With kernel-based classifiers such as SVM, the classification of an unseen instance is done by finding out whether the instance is more similar to the ones which are related by the given relation than the ones which are not. In early works, the kernel functions employed commonly make use of structured syntactic information, such as constituency-based [Zelenko et al., 2002; Zhou et al., 2007] and dependency-based [Culotta and Sorensen, 2004; Bunescu and Mooney, 2005] parse trees. Correspondingly, the similarity scores are usually computed with graph algorithms, such as counting the number of common subtrees [Zhou et al., 2007] and measuring the number of common properties on the shortest path between pairs of entities [Bunescu and Mooney, 2005]. Therefore, these methods naturally handle structured information well. To allow more types and forms of information to be incorporated, recent research also works on developing more complex kernel functions.

For example, the composite kernel in [Zhang et al., 2006] combines an entity kernel and a tree kernel through polynomial expansion, while the context-sensitive convolution tree kernel in [Zhou et al., 2010] is specifically designed for a rich semantic relation tree structure which integrates both syntactic and semantic information. While these complex kernels are able to outperform the feature-based modeling methods, they require substantial efforts to engineer and it is unclear how applicable they are for relation extraction problems of different settings or in other domains.

Similar to entity extraction, many approaches in these two categories are supervised and their effectiveness is dependent on the availability of an annotated corpus of suitable size. To alleviate this need and tap into the large amount of unlabeled data from large text collections or the Web, non-supervised approaches have also been an active area of research in relation extraction. For example, two early rule-based systems, DIPRE [Brin, 1999] and Snowball [Agichtein and Gravano, 2000], start with a seed collection of entity pairs for the relation to be extracted. They then search in unlabeled text sources (*e.g.*, the Web) for sentences containing the entity pairs. Afterwards, they learn new rules from the retrieved sentences and use the learned rules to extract new entity pairs from the text sources. These entity pairs are then added to the seed collection and the process repeats until some termination condition is met. Later systems, such as KnowItAll [Etzioni et al., 2005] and TextRunner [Banko et al., 2007], make use of generic patterns to extract candidate entity pairs. These candidate pairs are then selected using domain-independent heuristics (*e.g.*, pointwise mutual information derived from search engine hit counts) or unsupervised classifiers (*e.g.*, a classifier that heuristically labels its own training data). In the end, the selected pairs can be used to derive extraction patterns or provide statistics for estimating whether an entity pair is a correct instance. In the case where a relation database exists, distant supervision can be performed by harvesting training data using the entity pairs from the database [Mintz et al., 2009].

Moving beyond binary relations, rule-based approaches are more popular because they handle multi-way relations naturally by defining patterns over multiple entities and reporting that the relations of interest exist among the entities

matched. For example, [Aone and Ramos-Santacruz, 2000] extract 61 types of events using 50 generic event extraction patterns supported by lexico-syntactic information. These patterns can be learnt automatically (*e.g.*, [Piskorski et al., 2007]). As a way to consolidate texts that contain similar events for better rule learning and relation extraction, clustering can be applied as a preprocessing step [Piskorski et al., 2008; Liu et al., 2008].

In contrast, in statistical approaches, multi-way relations need to be decomposed to multiple binary relation classifications whose results need to be combined. [McDonald et al., 2005] propose to factorize the complex relations into a set of binary relations and train one classifier to extract all pairs of related entities. Based on the output of this classifier, a graph can be constructed with nodes representing entities and edges representing whether the entities are related. The original multi-way relation can then be recovered by finding the maximum cliques in this graph. The main advantage of this method is that it allows statistical approaches for binary relation classifications, which have been studied extensively, to be applied onto multi-way relations.

Research of domain-specific relation extraction has been done predominantly in the biomedical domain, for tasks such as gene-drug relation, protein-protein interaction and bio-molecular event extraction. In general, both rule-based [Hakenberg et al., 2008] and statistical approaches [Riedel and McCallum, 2011; Tikk et al., 2010] have been adopted equally, although the results in [Kim et al., 2011] give some evidence that the latter approach leads to better performing systems. Various domain-specific sources can be utilized in the extraction process. For example, medical information databases (*e.g.*, PharmGKB) can be used to perform distant supervision [Buyko et al., 2012] while lexica of problem-specific trigger words (*i.e.*, words that usually express interactions) can be used to avoid extracting relations from irrelevant sentences [Bobic et al., 2012].

All the existing works in relation extraction focus on extracting relations between two textual entities. As far as we know, no prior work has examined the extraction of relations between text entities and domain-specific constructs. Therefore, we carry out our own corpus study in math to get a better understanding of how concepts and constructs can be related and then formulate the

Table 5.1: Wikipedia pages used in corpus study.

Title	URL
Absolute value	http://en.wikipedia.org/wiki/Absolute_value
Bayes' theorem	http://en.wikipedia.org/wiki/Bayes'_theorem
Complex number	http://en.wikipedia.org/wiki/Complex_number
Fraction	http://en.wikipedia.org/wiki/Fraction_(mathematics)
Fourier transform	http://en.wikipedia.org/wiki/Fourier_transform
Function	http://en.wikipedia.org/wiki/Function_(mathematics)
Modular arithmetic	http://en.wikipedia.org/wiki/Modular_arithmetic
Polynomial	http://en.wikipedia.org/wiki/Polynomial
Pythagorean theorem	http://en.wikipedia.org/wiki/Pythagorean_theorem
Trigonometry	http://en.wikipedia.org/wiki/Trigonometry

problem of Text-to-Construct Linking accordingly.

5.1.2 Insights from Corpus Study

We perform our corpus study on a sample of math resources from our math corpus (as described in Section 4.3.1) from which we identify concepts and expressions as well as the possible types of relations between them.

We have randomly selected 10 Wikipedia pages from our corpus on 10 different math concepts, such as absolute value and Fourier transform. The complete list of the Wikipedia pages is presented in Table 5.1.

For each Wikipedia page, we identify the contained concepts and expressions through the follwing semi-automatic process:

- First, we automatically tokenize all the text (including the alternative text for images), assign POS tags to the tokens, and perform text chunking.
- Afterwards, for each phrase detected through the chunking process, we automatically check whether it contains a sub-phrase which appears in the math concept list (same as the one described in Section 4.3.1). If so, this phrase is marked as a candidate concept that can be linked to the expressions. For example, after chunking, noun phrases such as "a degree 0 polynomial" and "ancient history" may be detected. Since "polynomial" appears in the math concept list while neither "history" nor "ancient his-

Table 5.2: Semantic relations between concepts and expressions.

Name	Definition	Example	Count
Representation	The expression denotes the math representation of the concept.	**A complex number** is a number which can be put in the form $z = a + bi$.	906 (59%)
Property	The property of the expression is specified by the concept.	For **any real numbers** x and y, ...	294 (19%)
Argument	The expression serves as the argument of the concept.	**Divide** *3* by *4*... **Substitute** y with $x^2 + 1$...	50 (3%)
Context	The expression sets the context of the concept.	**The absolute value** of x...	176 (11%)
Co-reference	The expression is referred to by the concept.	...$3^2 + 4^2 = 5^2$. **The previous equation**...	128 (8%)

tory" does, we mark the former as a candidate concept but not the latter.

- We then automatically mark all the non-word and non-punctuation text tokens, as well as the LaTeX expressions in the alternative texts of the expression images on the pages as math expressions.

- Lastly, we manually go through the pages to identify the concepts and expressions that have been missed in the automatic marking process and correct the errors in marking as necessary.

An example of the identified concepts and expressions on a page is as follows: (The concepts are in **Bold** while the expressions are in *italics*.)

> If we let c be **the length** of **the hypotenuse** and a and b be **the lengths** of **the other two sides**, **the theorem** can be expressed as **the equation**: $a^2 + b^2 = c^2$.

In total, we have identified 8,121 concepts and 2,434 expressions from the selected pages.

After the identification step, we examine how the concepts are semantically related (*i.e.*, linked) to the expressions, when applicable.

In our domain study of math, we have coded five distinct types of semantic relations altogether, as summarized in Table 5.2.

Among these five types of semantic relations, we find the *representation relation* most important for domain-specific IR. It can be used to resolve concepts to their representations and implement the features mentioned at the beginning of this chapter. Therefore, the extraction of this relation is the focus of this chapter and forms the basis of the problem of Text-to-Construct Linking.

The other relations are not directly relevant to domain-specific IR; however, they are still useful in their own ways in other contexts. For example, they can be useful in document understanding and expression analysis: The *property relation* keeps track of the properties of the variables (*e.g.*, whether a particular variable is positive/negative). When these variables are used later in some other expressions, these properties may serve as descriptions to individual variables for the users or clues for deciding whether two variables are of the same nature during indexing. As another example, while the *argument relation* is not very common, it provides information about how one expression can be transformed to another. By consolidating and analyzing such information, we will be able to know whether two seemingly different expressions are equivalent (up to a few steps of transformation) or indeed different. Such knowledge would allow the search systems to cluster related expressions together during indexing and improve the recall of retrieval. The *context relation* may seem uninformative in isolation, but when coupled with the representation relation, can assist in connecting related expressions. For example, in the sentence "**The absolute value** of x is denoted as $|x|$.", we would be able to correctly establish the fact that $|x|$ is related to x as a way to express its absolute value through the context relation "The absolute value $\leftrightarrow x$" and the representation relation "The absolute value $\leftrightarrow |x|$". Last but not least, the main objective of the *co-reference relation* is not to relate a concept to an expression or vice versa. Instead, it is meant to introduce an expression into another part of a resource so that more relations can be established for it. Therefore, the detection of this relation can be done as a preprocessing step to facilitate the detection of other relations.

Aside from identifying and coding the possible semantic relation types, we have also surveyed our dataset for two sets of statistics to characterize the nature of the representation relation.

Table 5.3: Multiplicity of the representation relation.

	To None	To One	To Many
Concept	7,368 (91%)	652 (8%)	105 (1%)
Expression	1,554 (64%)	854 (35%)	27 (1%)

Table 5.4: Distance between related concepts and constructs.

Adjacent	One to three words apart	Four or more words apart
396 (45%)	300 (34%)	189 (21%)

The first statistic collected we term *multiplicity*, which specifies how many expressions are related to one concept in a sentence through the representation relation and vice versa.

As shown in Table 5.3, most of the concepts (91%) are not related to any constructs through the representation relation. This is expected since concepts are often mentioned in text without their representations in expressions. In contrast, more than one third of math expressions are related to exactly one concept. This indicates, whenever an expression appears, there is a good chance that the concept it represents can be found in the same sentence. Moreover, it is possible (although unlikely) for one expression to be related to multiple concepts. This happens when multiple names are introduced as different ways to call the same expression.

We have also analyzed representation *distance*, which measures how far two related concept and construct are apart from each other, in number of words.

As shown in Table 5.4, when a concept is related to an expression through the representation relation, they are often (79% of the time) adjacent or within one to three words apart. Given the close proximity of a concept and its related expression, it is likely that distance information is useful in extracting the representation relation. Note that we do not consider a concept and an expression to be related by the representation relation if they are not in the same sentence. In such cases, a text phrase would have been used to introduce the concept or the expression into the sentence of the other. Therefore, co-reference resolution is required to resolve the text phrase to the concept or expression it refers to before relation extraction can be performed on the resulting pair of concept and

expression in the same sentence.

5.2 Problem Formulation

Based on the findings from our corpus study, we formulate the problem of Text-to-Construct Linking as follows:

Given a set of domain-specific resources (*e.g.*, math webpages), in which domain-specific concepts (*e.g.*, math concepts) and constructs (*e.g.*, expressions) have been identified, for each identified concept, return a ranked list of constructs which are the possible representations of this concept.

By formulating the problem in this way, we have scoped certain concerns out of our research on Text-To-Construct Linking while including some others as explained below:

With respect to limitations, we assume that the concepts and constructs are already identified in the resources. Both identification tasks are in fact instances of Resource Categorization on nominal facets at sub-segment level (*i.e.*, classifying words/symbols based on whether they are part of a domain-specific concept or construct). Therefore, as demonstrated in Chapter 3, these tasks can be done through supervised learning and their correlations with categorizations at segment-level (*e.g.*, sentence-level) can be exploited as needed for better performance. Alternatively, if a list of domain-specific concepts is available, the concept identification task may also be done through word matching. Therefore, we believe these tasks can be performed adequately with existing approaches and hence they can be scoped out from this part of our research.

In addition, as mentioned at the beginning of the chapter, it is not uncommon for a concept to have multiple representations. Some may be due the context in which the concept is discussed (*e.g.*, complex numbers represented in Cartesian form or Polar form), while some others due to the level of details needed (*e.g.*, a polynomial denoted as one single symbol or a sum of terms). These representations are not necessarily equally informative to users or useful in expression retrieval. Therefore, the constructs for the same concept need to be ranked so that the more informative/useful ones can be selected and utilized.

5.3 Methodology

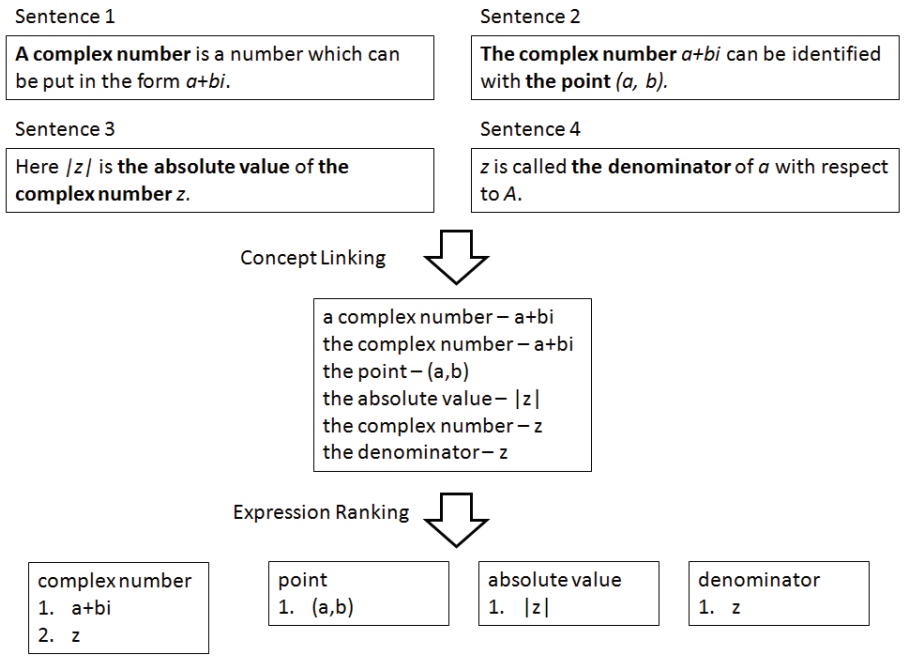
We address Text-to-Construct Linking in the domain of math in two stages:

The first stage is the concept linking stage. In this stage, we determine whether a pair of concept and expression is related by the representation relation. This is done by supervised learning.

The second stage is the construct ranking stage. In this stage, all the concept-expression pairs identified in the previous stage are consolidated by concepts. A TF.IDF-like utility score is then computed for each expression related to a particular concept. In the end, all the expressions are ranked based on this utility score to produce an ordered list of expressions for each concept.

An illustrated example of this process can be found in Figure 5.2.

Figure 5.2: Example of Text-to-Construct Linking in math.



5.3.1 Concept Linking

Given a collection of resources with concepts and expressions identified, we process each of the resources in turn. For each pair of concept and expression within the same sentence, we link them up if they are in a representation relation. This process is summarized in Algorithm 5.1.

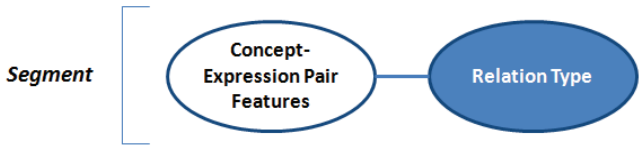
Algorithm 5.1 link-concept(*resources*)

```
1: create pair list pairList
2: for each resource r ∈ resources do
3:   for each sentence s ∈ r.sentences do
4:     for each concept c ∈ s.concepts do
5:       for each expression e ∈ s.expressions do
6:         if is–linked(c,e) then
7:           create pair pair
8:           pair.concept = c
9:           pair.expression = e
10:          add pair to pairList
11: return pairList
```

The key component for this algorithm is the *is-linked* function in Line 5. Given a concept and an expression, this function determines whether the representation relation exists between them.

In our correlation graph, this binary decision is represented by the relation type node at the sub-segment (*i.e.*, concept/expression) level. As shown in Figure 5.3, this decision can be made by exploiting the correlation between the relation type node and its observable characteristic nodes.

Figure 5.3: Correlation exploited for Text-to-Construct Linking.



We propose to implement this function using supervised learning. Since only the representation relation is considered, it is effectively a binary classification problem and different sources of information (including distance information) can be incorporated through features.

In total, four sources of information (listed in Table 5.5 as feature groups) are considered. The first source describes the given concept and expression in a textual, non-domain-specific manner. For the concept, we put in the standard text features, such as n-grams, head word and length. For the expression, we also compute its n-gram features based on its symbols and put in a feature for its length. We incorporate information about how the concept and the expression are relative to each other through a second feature group called rel-

Table 5.5: Feature groups for concept linking.

Group	Definition	Examples
Concept & Expression	Features that describe the concept and the expression individually.	n-grams (sequences of n words, where $1 \leq n \leq 3$), head word (for concept), length.
Relative Information	Features that describe how the concept and the expression are relative to each other.	Distance, relative position, in-between n-grams.
Preferential Information	Features that describe a concept (or expression) which is closer to the expression (or concept) being examined.	Existence, relative position, and in-between b-grams of a concept (or expression) which is closer to the expression (or concept) being examined.
Domain-specific Knowledge	Features that describe the concept and the expression, as well as the constraints between them and the context around them, in a domain-specific manner.	Type of concept and expression, selection restriction, domain-specific text cues, expression semantics.

ative information. Besides the distance, the relative position (*i.e.*, whether the concept is before or after the expression) and the n-grams of the text between them also belong to this group. The third group, preferential information, captures whether other concepts (or expressions) are closer to the target expression (or concept) currently being examined. This source of information is included because a nearer (in words) concept (or expression) is likely to be preferred given the fact that linked concept-expression pairs are usually very close to each other. The fourth group contains features derived from domain-specific knowledge, such as whether the types/meanings of the concept and expression coincide and whether the concept-expression pair fits particular writing patterns in math resources. These features are intended to provide more accurate representations of the concept-expression pair, model the constraints between them and capture their contexts in a domain-specific manner. By introducing domain-specific knowledge, we hope to assess whether it is worthwhile to utilize such features.

We evaluate this approach and perform a detailed analysis in Section 5.4.

5.3.2 Construct Ranking

Once the related concepts have been identified for each expression, we proceed to the construct ranking stage. This stage is divided into three steps as summarized in Algorithm 5.2.

The first step is to consolidate the expressions by related concepts. This is done by grouping the concepts which share the same longest sub-phrase as found in the math concept list. For example, "some polynomials" and "this polynomial" will be grouped together since they share the same longest sub-phrase of length 1 ("polynomial"); however, they will not be grouped with "zero polynomial" since the latter by itself is a sub-phrase of length 2 that can be found in the math concept list. Afterwards, all the expressions related to at least one of the concepts in a group will be consolidated as a list of related expressions for this group of concepts. The advantage of grouping concepts in this way is that each concept group would be able to cover possible lexical variations of the same math concept, while all the subtypes of this math concept will have their own groups such that a search on the subtypes would not incorrectly retrieve expressions related to the original math concept.

After consolidating the expressions, in the second step, we compute a utility score for each expression exp in a particular list l based on its TF.IDF as follows:

$$utility_{exp,l} = \frac{freq(exp,l)}{\sum_{list \in List} occur(exp,list)}, \tag{5.1}$$

where $freq(exp,l)$ is the frequency of the expression exp in the list l and $occur(exp,list)$ is a binary function indicating whether the expression exp occurs in the list $list$.

The key idea for this utility score is that, if an expression frequently appears as the representation for a concept group but much less so (or never) for other groups, then this expression should be a commonly-accepted representation unique to the concept group and hence a suitable candidate to be presented to the user and used for further expression retrieval.

In the final step, we rank the expressions in the same list based on their utility scores. For each concept identified in the given collection of domain-specific resources, the ranked list of related constructs is then the one associated

to the group which the concept belongs to.

Algorithm 5.2 rank-expression(*conceptList*, *pairList*)

```
1: create group list groupList
2: for each pair p ∈ pairList do
3:    group = get–group(p.concept, conceptList, groupList)
4:    add p.concept to group.concepts
5:    add p.expression to group.expressions
6: for each group g ∈ groupList do
7:    for each expression e ∈ g.expressions do
8:       e.utility = utility(e, g, groupList)
9: for each group g ∈ groupList do
10:   sort–by–utility(g.expressions)
11: return groupList
```

5.4 Evaluation

There are three main objectives for our evaluation: 1) to examine the utility of generic approaches, 2) to assess the need for domain knowledge, and 3) to identify the key challenges in Text-to-Construct Linking.

With these objectives in mind, for the concept linking stage, we evaluate our approach using our annotated corpus, compare the differences in performance due to the use of different groups of features, and perform a detailed error analysis on the classification results. As for the construct ranking stage, we opt for a qualitative analysis instead of a quantitative one due to the limited size of our annotated corpus. Nevertheless, our manual inspection on the ranking results has also yielded interesting findings in accordance with our objectives.

5.4.1 Concept Linking

We evaluate our approach for the linking function using all the concept-expression pairs within the same sentence in our annotated corpus. We consider the concept-expression pairs with representation relation (based on our annotation) as positive examples and the rest as negative examples. We then train and test a CRF classifier on this set of data using 5-fold cross validation[1] to obtain the preliminarily results. We optimize the supervised learning approach by manually selecting the features that contribute positively to performance (See Table 5.6

[1]The same for other experiments in this subsection that involve supervised learning.

Table 5.6: Selected and rejected features for each feature group.

Group	Selected Features	Rejected Features
Concept & Expression	Concept, concept. n-grams, concept head, expression symbol	Concept length, expression n-grams and expression length.
Relative Information	Distance, relative position, n-grams in between, words before.	Words after.
Preferential Information	Existence, relative position of a closer concept and the n-grams between the expression and this closer concept.	Existence, relative position of a closer expression and the n-grams between the concept and this closer expression.
Domain-specific Knowledge	Type of concept and expression, domain-specific text cues, patterns of concept and expression types, selection restrictions.	

Table 5.7: Evaluation results on concept linking.

	P	R	F
Heuristics			
Distance	.36	.77	.49
Supervised learning (CRF)			
Concept and Expression only	.32	.36	.34
(up to) Relative Information	.81	.80	.80
(up to) Preferential Information	.82	.81	.81
All feature groups	.84	.81	.82
Supervised learning (linear-kernel SVM)			
All feature groups	.81	.80	.80
Supervised learning (RBF-kernel SVM)			
All feature groups	.85	.84	.84

for the list of selected and rejected features). In addition, to examine the utility of individual feature groups, we also introduce the feature groups one by one to measure their impact on performance. Last but not least, since kernel functions have been shown to be useful in achieving good performance, we have also experimented with two automatic kernel methods: linear and RBF kernel SVMs[2]. The evaluation results, as measured by the standard IR metrics of precision, recall and F_1-measure on the positive class, are as shown in Table 5.7. For comparison, the performance of a simple heuristic-based approach, which considers a concept and an expression to be linked if they are no more than three words apart, is also listed.

The performance of the distance heuristic baseline garners 0.49 on F_1-measure. This is a plausible result as it only takes distance information into account. Nev-

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

ertheless, the low precision (0.36) and high recall (0.77) of this approach reflects the fact that, while most linked concepts can be found within three words from an expression, not all the concepts in this range are related to it.

In comparison, the supervised learning approaches (with all feature groups) perform much better ($\geq$ 0.80 on all metrics). By taking into account various sources of information, they are able to detect the representation relation with reasonable precision without sacrificing recall.

Among the three supervised approaches, RBF-kernel SVM slightly outperforms CRF (+0.01, +0.03 and +0.02 on the three metrics), which in turn slightly outperforms linear-kernel SVM (+0.03, +0.01 and +0.02). This indicates that a suitable kernel does have a positive effect on the extraction performance.

In terms of the contributions of individual feature groups, using only the features from the concept & expression group results in worse performance than distance-based heuristics. This confirms that distance information is rather important for this problem. Moreover, the selected features are mostly about the concept and the only feature selected for expression is the symbol feature. In other words, the structure of the expression, as represented by n-grams, do not play a key role in determining whether it is linked to a concept.

With the inclusion of relative information, the performance of the supervised learning approach improves significantly, close to the performance with all feature groups included. Among the three features selected for this group, the distance feature contributes the most, followed by the in-between n-grams. This is rather intuitive since the text between a concept and an expression (if any) is usually indicative of the relation between them.

The contributions from the preferential information and domain knowledge feature groups are minor. In the feature selection process, we have discovered that it is useful to know whether a closer concept exists but not a closer expression. In addition, some domain knowledge features, such as the type of an expression (*e.g.*, number/variable), are also useful; however, it takes more time to come up with and test these features than the ones in other feature groups and the coverage of these features is very limited (*i.e.*, only triggered for very few pairs). Therefore, we find it not cost-effective to try to improve the accuracy

through developing features based on domain knowledge.

To further understand the behavior of the learned models, we have performed an error analysis on the linking outputs and made the following observations:

First of all, depending on the structure of the sentence, an expression may be distant from its related concept due to the existence of additional syntactic components. For example, in the sentence "$f_Y(y|X = x) = L(x|y)$ is, as **a function** of x, **the likelihood function** of x given $Y = y$", the concept "**the likelihood function**" should be linked to the expression "$f_Y(y|X = x) = L(x|y)$". However, the intervening syntactic clause "as **a function** of x" inflates the distance between them and creates a long-distance dependency that confuses the classifier. Syntactic information is needed to correctly gauge the distance between them; however, parser accuracy may be compromised due to the presence of math expressions. Therefore, domain-specific parsers are needed to obtain reliable syntactic information for this problem.

Second, when coordinating conjunctions (*e.g.*, "and" and "or") are used to relate multiple concepts to one expression or vice versa, the classifier is usually able to link the pair whose distance is the smallest but misses the others. For instance, in the sentence "For **any real number** a, **the absolute value** or **the modulus** of a is denoted by $|a|$.", the link between the concept **the modulus** and the expression $|a|$ is often identifiable by the classifier but not the one between **the absolute value** and $|a|$. To address this type of error, a preprocessing step may be performed to identify these concepts/expressions and treat those that are connected by the same coordinating conjunctions as one single super concept/expression in the linking process. After the linking process, all the links detected for the super concept/expression can then be assigned to all its constituting members.

Last, another challenge for the problem is the lexical/syntactic variations of the representation relation. In math resources, the concept can be mentioned first before introducing the corresponding concept or vice versa. In both cases, there are many possible wordings. To name but a few, for the first case, there are "[Concept] is denoted as [Expression]", "[Concept] is given by [Expression]" and "[Concept] can be written as [Expression]", while for the second case, there

Table 5.8: Examples of rankings produced for groups of concepts.

Concept group	Score	Expression
Distance, the Euclidean distance, the distance, the standard Euclidean distance	2.0	$\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$
	1.0	$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}$
	0.25	c
	0.14	b
	0.12	a
Absolute value, absolute Values, an absolute value, the absolute value	2.0	$\|z\|$
	1.0	$\|a\| = \begin{cases} a, & \text{if} a \geq 0 \\ -a, & \text{if} a < 0, \end{cases}$
	0.33	$r = \|z\| = 0$
	0.14	3
Conditional probability, conditional probabilities	1.0	$P(A\|B)$
	0.5	$1320 \int_{1/2}^{1} r^7(1-r)^3\, dr \approx 0.887$, $P(B\|A)$

are "[Expression] is called [Concept]", "[Expression] represents [Concept]", and "[Expression] denotes [Concept]". As discussed in Section 3.5 for key information extraction, we may also need to employ more sophisticated statistical models to manage these variations.

To sum up, in our current study, supervised learning works well for the concept linking stage. Encoding a certain degree of domain knowledge helps to improve performance, but in this study proved to be cost-ineffective.

5.4.2 Construct Ranking

In our annotated corpus, there are 212 groups of concepts with at least one linked expression. A few examples of such groups and their linked expressions can be found in Table 5.8. On average, the number of unique expressions linked to each group is 3.16. Due to this limitation in data size, our evaluation on construct ranking is more qualitative in nature. We have examined these concept groups and their linked expressions one by one and made the following observations on the ranking process:

First of all, the grouping of concepts by longest matched entry in the concept list generally works as intended. For example, the group on polynomials consists of mostly textual variants of the concept, such as "a given polynomial", "each polynomial" and "such polynomials", while special types of polynomials, such as "zero polynomials" as listed in the concept list, are in their own groups.

Nevertheless, we have also observed that certain specific subtypes of a common concept, such as "Short-time Fourier transform" of "Fourier transform", are missing from our concept list and hence not separated into their own groups. In our opinion, this can be addressed by merging (*i.e.*, computing the union of) multiple concept lists from different sources so that the coverage of the resulting concept list is ensured.

Secondly, expression-TF.IDF is effective in demoting generic representations (*e.g.*, variables) of concepts to the bottom of the list while promoting more specific ones (*e.g.*, formula) to the top. This is best observed in the concepts which have representations in both formula and variables. For example, in the ranked list produced for "distance", the formula representation $\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$ appears at the top followed by a few other variations, and at the bottom, the single variable representations such as a and b. In our opinion, such generic representations give little information about the concepts and may lead to the retrieval of irrelevant materials since they can be used to refer to other concepts as well. In contrast, the formula representations, which give detailed descriptions of the concepts, are more informative to users and the resources containing them are likely to be discussing about the same concepts due to their specificity. Therefore, we believe the ranking produced by expression-TF.IDF helps to find suitable expressions for display and expression retrieval.

Nevertheless, besides TF.IDF, other factors may also come into play in the ranking process. For example, if users are looking for actual instances of concepts for intuitive impressions or exact values for calculation, concrete expressions, such as $2 + 3i$ for complex numbers or 3.14159 for the constant Pi, may be more suitable than abstract ones, such as $a + bi$ or π. In this case, the concreteness of an expression matters. As another example, users with little background may prefer simpler expression representations (*e.g.*, triangle area as $\frac{1}{2} * b * h$), while experts are able to understand and benefit from more complex ones (*e.g.*, triangle area as Heron's formula $\sqrt{s(s-a)(s-b)(s-c)}$). To cater for this difference in background, the complexity of expressions needs to be taken into account. Therefore, in practice, there may be a need to create separate expression lists with different ranking criteria for different scenarios.

Lastly, as far as our research has progressed, deep domain knowledge does not seem vital for construct ranking, although some shallow domain knowledge may be useful. For instance, the computation of expression-TF.IDF would be more accurate if we know how to normalize expressions written in different forms. Similarly, the computation of expression concreteness and complexity may benefit from knowing the basic types of math symbols (*e.g.*, numbers, variables and operators) and how they can be put together to form expressions. Nevertheless, unlike the matching process in expression retrieval, the ranking process' purpose is primarily to establish a relative order, and does not need to be highly precise. It is not mandatory to acquire much domain knowledge for this process.

5.5 Future Work

Due to manpower and time constraints, the corpus study is only done with a small set of Wikipedia pages. Although these pages are of good quality, contain abundant semantic relations, and provide textual forms for all the math expressions, they only represent one form of domain-specific resources (encyclopedic, targeted at intermediate to expert readers). To capture other textual variations of the existing semantic relations and discover new ones, we need to include resources of other types (*e.g.*, research articles) or targeted at different audience (*e.g.*, elementary webpages). For this purpose, we can sample and analyze resources from publicly accessible paper databases (*e.g.*, ArXiv) and well-curated educational websites (*e.g.*, cut-the-knot.org).

In addition, as mentioned in the corpus study, other semantic relations can improve domain-specific search systems – from backend (better indexing & retrieval of domain-specific constructs) to frontend (additional sources of information for the user interface). While intentionally excluded from our current research, the detection and utilization of these relations is certainly a promising direction for future research. As a general approach, we can find a suitable feature set for each relation and apply supervised learning as is done for the representation relation. For the argument and co-reference relations, we may borrow techniques from semantic role labeling and co-reference resolution from natural

language processing research, since the detection of such relations is essentially the domain-specific version of these two problems. In terms of utilization, we can replace the referring texts with constructs based on co-reference relations to improve the detection performance or relate constructs through concepts based on context relations. In contrast, the utilization of the property and argument relations is less straightforward and commonly requires some degree of construct analysis. For example, we need to be able to determine whether a construct occurs in another, so that we can 1) take the identified properties of the former construct into consideration when analyzing or presenting the latter, or 2) understand which part of the latter construct is affected by the transformation. The utilization of these two relations is best studied as specialized research on the constructs in the corresponding domain.

5.6 Discussion

The idea of Text-to-Construct Linking is to automatically identify the semantic relations between domain-specific concepts and constructs, and make use of such relations to facilitate domain-specific IR. Among the semantic relations identified from our corpus study in math, we believe the representation relation (relating a concept with its representation in constructs) is the most important since it allows users to stick to keyword search while the search system incorporates relevant domain-specific constructs in the retrieval process.

Using math expressions, which is an example of inline, text-based constructs, we have shown that the representation relation can be extracted using a supervised learning approach and suitable expressions can then be selected automatically using simple heuristics for display and expression retrieval.

Since we do not make any assumption on sentence structures or rely on deep domain knowledge about the constructs, we believe our method for relation extraction is portable to other domains as long as the constructs are written as part of sentences. When porting our approach to other domains (or onto a different set of math resources), we can also start with features that do not rely on domain knowledge. After their utility have been exhausted, we can then conduct corpus

studies to find out whether domain knowledge can taken into consideration in a cost-effective way and design relevant features as necessary. In addition, as a caveat of supervised learning in all forms, the cost and difficulty in obtaining suitable annotations may limit the portability and scalability of our approach. In our opinion, if it is still possible to obtain a small set of high quality annotations, then bootstrapping with this set of annotations can be a viable option. Otherwise, we can make use of some simple heuristics/rules to gather training data. For example, we can consider a pair of concept and expression to be linked if they are adjacent to each other or follow the lexical pattern "[Concept] is denoted as [Expression]". This set of training data (with some manual selection if possible) can then serve as the starting point for bootstrapping.

In the case where the constructs are not inline (*e.g.*, stand-alone math expressions) and/or graphical (*e.g.*, structural formula in chemistry), additional preprocessing needs to be done to identify the mentions of the constructs in text. The difficulty of this preprocessing step depends on whether the constructs are referred to by labels or free text. The former case is straightforward since the labels are unique and they usually follow some fixed style within a resource. In contrast, the latter case is more tricky due to the possible variations of the phrases used to make the reference. Nevertheless, once such mentions have been identified, the referred constructs can be considered as inline (*i.e.*, appearing at the locations of the mentions) and treated accordingly. As an alternative, it is also possible to treat such constructs as segments and identify other segments (*e.g.*, paragraphs or sentences) that are semantically related to them. This can be done using a supervised learning approach similar to our current approach. Afterwards, concepts can be derived from the identified segments (*e.g.*, using heuristic selection or topic modeling) as the ones linked to the constructs.

As for construct ranking, our current utility score is computed based on construct-TF.IDF and hence it requires a way to decide whether two constructs are the same such that the frequency of the constructs can be computed. This is not a strong requirement since any domain-specific search system with construct retrieval capabilities should have a function to compute the similarity between two constructs. Based on this similarity function, we can then set a threshold

such that any two constructs with a similarity higher than this threshold are considered to be the same. In this way, construct-TF.IDF can be computed and construct ranking can be performed as described above. Therefore, we consider our approach for construct ranking to be portable to other domains as well. Nevertheless, in the case where the similarity between the constructs cannot be computed, construct-specific metrics (*e.g.*, concreteness and complexity) or external information (*e.g.*, quality of resources) can be used instead to produce a construct ranking.

Chapter 6

Integrating Domain-specific Components into IR Applications

To demonstrate the applicability and usefulness of our research, we have implemented two domain-specific search systems, one in the domain of math and the other in healthcare. Both systems incorporate and extend the features described in the previous chapters to handle domain-specific user needs. The math system incorporates features based on Resource Categorization at resource-level and sentence-level, as well as Text-to-Construct (*i.e.*, Text-to-Expression) Linking. As shown in our evaluation, this system is significantly more effective for math search than a similar system without the aforementioned features. As for the healthcare system, it also performs categorization at multiple levels (*i.e.*, resource-level, sentence-level and word-level). While Text-to-Construct Linking is not applicable in healthcare, this system is equipped with additional features for better workflow integration.

The rest of this chapter is organized as follows. We detail our math system and the evaluation we have performed on it in Section 6.1 and Section 6.2 respectively. We then move on to describe our healthcare system in Section 6.3.

6.1 Math Search System

The design of our math search system is directly motivated by our user study with math seekers. As summarized in Section 2.2, they often turn to web searches to fulfill their resource and information needs about math concepts, as it is quick, convenient and able to provide a variety of resources and information. However,

the lack of organization of results by resource type, information type and audience level often drives them to use specialized search engines. Such specialized search engines are better equipped to handle domain-specific organization, but less convenient and less accessible. Importantly, the fact that direct expression search has not been well-received indicates that there is a usability gap between the input mechanism (*i.e.*, keyword search) and expression retrieval.

Therefore, the central idea of our search system is to improve the organization of results using Resource Categorization at resource-level and sentence-level. We also enable expression retrieval while retaining keyword search as the main input mechanism. This is done by deprecating direct expression input by using Text-to-Construct Linking to allow expression retrieval by standard keyword search.

The resulting system has the following two key features:

Feature 1: Automated categorization of resource type, information type and readability. Our system automatically categorizes the resources into predefined types and their component sentences into information types. It also computes the readability of the resources and categorizes them accordingly on a discrete 5-point scale. Such information is displayed in the search results and can be used for filtering and sorting. This feature allows users to focus on the resources that not only satisfy their needs but also are suited to their math background.

Feature 2: Automated linking of keywords to their expression representations. Our system keeps a list of math concepts and their expression representations discovered from the resource collection through automated linking. The expressions are presented to users and can be used for expression retrieval whenever the corresponding math concept is used as the keyword for searching. In this way, users can immediately be informed about the expression representations of the keyword and search with them without having to enter them manually.

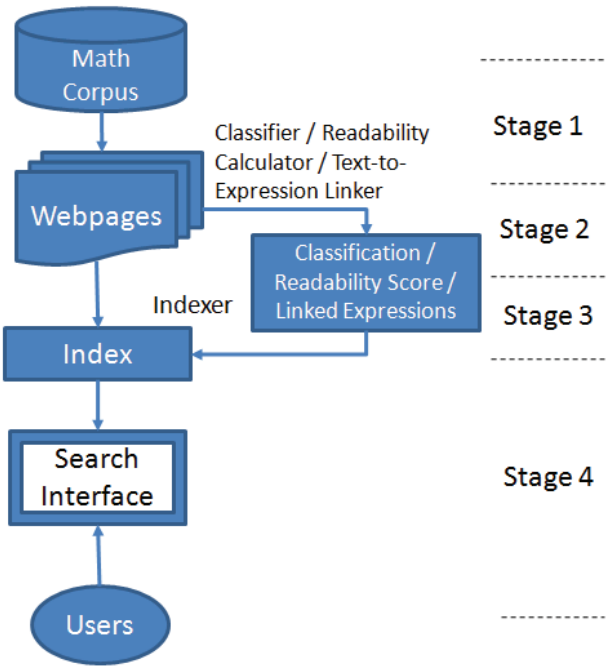
6.1.1 System Description

We now describe the architecture of our system and then explain the two key features in detail.

The architecture of our system (Figure 6.1) consists of four stages:

Stage 1 We take all the webpages from our math corpus (2,377 in total, as

Figure 6.1: Architecture of the math search system.



described in Section 4.3.1) as the resource collection for our system. This collection can be replaced by periodic crawling later if the system is scaled up for public use in future; however, for now, we use our math corpus as a convenient collection since it already contains a variety of math resources for the 27 chosen topics.

Stage 2 We then apply our machine-learned classifiers to categorize the resources, compute their readability using our iterative approach, and link math concepts to their expression representations.

Stage 3 We employ Lucene[1], a freely-available text search engine library, to index[2] the resources with the results from categorization and linking.

Stage 4 Users access the organized resources through our search interface.

Feature 1: Automated categorization of resource type, information type and readability. The webpages in the collection are of various types. To facilitate filtering in the downstream user interface, we apply supervised learning

[1] http://lucene.apache.org/

[2] Normalization and stemming are done via the StandardAnalyzer class. Same for all other systems mentioned in this chapter.

Table 6.1: Math resource types for classification.

Resource Type	Definition
Concept Information	Explanatory texts on math concepts.
Exercises	Exercises on math concepts.
Discussion	Forum discussions on math concepts.
Paper	Scholarly articles that describe research on math concepts.
Visualization	Applets, figures and diagrams that visualize aspects of math concepts.
Textbook	Textbooks on math concepts.
Tool	Software packages that facilitate the application of math concepts.
Course	Courses on math concepts.
Journal	Journals on math concepts.
Research Community	Events, conferences and researchers related to the research on math concepts.
Hub	Compiled links to resources on math concepts.
Others	Any other types of resources.

to classify the webpages into one of the 12 categories as listed in Table 6.1. Although by no means exhaustive, these categories are designed to meet the common resource needs of math seekers as discovered in our user study (See Table 2.1 for a list of resource needs).

We have annotated 1,068 webpages from our math corpus (*i.e.*, all webpages for 10 concepts and the ones in the top 30 search results for the remaining 17 concepts; discarding irrelevant webpages) as our training data. We then extract three classes of features from the webpages: token (*e.g.*, n-grams), webpage (*e.g.*, URL tokens and content length) and formatting (*e.g.*, whether a word is in bold/italics). Since we were able to clearly associate a single resource type to each webpage, we train a multi-class CRF classifier on our annotated data instead of multiple one-against-all classifiers. We then apply the resulting classifier onto the remaining webpages to determine their resource type.

The resource type of the search results are displayed together with the context of the matched keywords in the search interface. Users can also use the resource type filter to filter results to a specific type.

Among different types of resources, concept information resources commonly contain the most types of information (*e.g.*, definitions, exercises, examples and

Table 6.2: Math information types for classification.

Information Type	Definition
Definition	Sentences that contain definitions of math concepts.
Exercises	Sentences that contain exercises on math concepts.
Examples	Sentences that contain examples on math concepts.
Proof	Sentences that contain proofs on math concepts.
Others	Any other types of sentences.

proofs) sought by math seekers[3]. To save their trouble of reading through the resources to locate such information, we further categorize the sentences of the concept information resources into five categories as listed in Table 6.2.

The categorization process is similar to that of resource type. We have annotated all the sentences in the 112 concept information webpages on Bayes' theorem, complex numbers and modular arithmetic as our training data. We then extract five classes of features from the sentences: token (*e.g.*, n-grams), sentence (*e.g.*, length and position of the sentence), formatting (*e.g.*, whether a word is in bold/italic), concepts (*e.g.*, appearances of math concepts), and expressions (*e.g.*, appearances of math expressions). We also train a hard, multi-class CRF classifier for this categorization on our annotated data and apply it onto all the sentences in the webpages which have been categorized as concept information resources by the resource type classifier[4].

The sentences belonging to the first four information types can be viewed directly in the search results. Filtering on information type is also available for users to focus on sentences containing specific types of information.

Last but not least, math resources targeted at different audience are written at different levels of readability. To help users pick out the ones that are suited to their level of knowledge, we compute the readability scores for the resources as described in Chapter 4. The final scores are used directly for sorting and converted into a discrete 5-point scale for display and filtering.

Feature 2: Automated linking of keywords to their expression representations. We link math concepts to their expression representations as

[3]See Table 2.1 for the list of information needs.

[4]Unlike our work mentioned in Chapter 3, we do not employ joint inference to combine this categorization with resource type categorization. The reason is that the categorization of other types of resources would not benefit from information type categorization or vice versa since we only target the information from the concept information resources.

described in Chapter 5. When users perform keyword searches in our system, the top five linked expressions are displayed. They can then choose any of the linked expressions for expression retrieval.

Figure 6.2 shows the search interface of our system. It demonstrates how Resource Categorization and Text-to-Construct Linking are employed: The categorization labels – resource type, information type and readability – are shown as part of the search results. Tools for filtering and sorting are provided on the left of the interface. The linked expressions are displayed above the search results with checkboxes for users to choose which one(s) to search with.

6.2 Evaluation for the Math Search System

To evaluate whether all these features in our system are effective in helping math seekers in their searches, we have conducted a system evaluation in which participants are required to use our system or a baseline to perform some math-related search tasks and share their opinions about these two systems.

We have carried out two versions of the evaluation with two different groups of participants: a face-to-face version with students from National University of Singapore (NUS), and an online version with Amazon Mechanical Turk[5] (Mturk) workers. The face-to-face version is qualitative in nature and similar to our earlier user study. In this version, we observe participants' actual search process *in situ* and get them to share their opinions through a semi-structured interview. In contrast, the online version is quantitative in nature. We put the two systems online for the Mturk workers to use to complete the search tasks. After completing the tasks, the workers are required to fill in a survey which contains the essential questions from the semi-structured interview. A summary of the steps in both versions of the evaluation can be found in Figure 6.3. By having both versions, we are able to obtain both qualitative and quantitative evaluation results for our system.

We have prepared four search tasks (listed in Table 6.3). Each task pictures a common scenario which calls for a search for math webpages for information

[5] https://www.mturk.com/mturk/welcome

Figure 6.2: Search interface of the math search system showing easy concept information resources containing definitions for complex numbers, with the linked expression $a + bi$ displayed and used for expression retrieval.

Enter your query here:

(Use commas to separate the keywords in your query. Example: right triangle, geometry.)

complex numbers [Search]

Type:
All (67)
Concept Information (52)
Exercise (3)
Discussion (2)
Tool (7)
Miscellaneous (3)

Difficulty:
All (74)
Very Easy (1)
Easy (52)
Intermediate (16)
Difficult (5)

Sort by:
Most Relevant First
Easiest First

Contains:
Definition
Exercise
Example
Proof

Hits **1-10** (out of about 16 total matching pages):

Representations of **complex numbers:**

- $a + bi$
- z
- $re^{i\theta}$
- $r(cos(t) + isin(t))$
- (a,b)

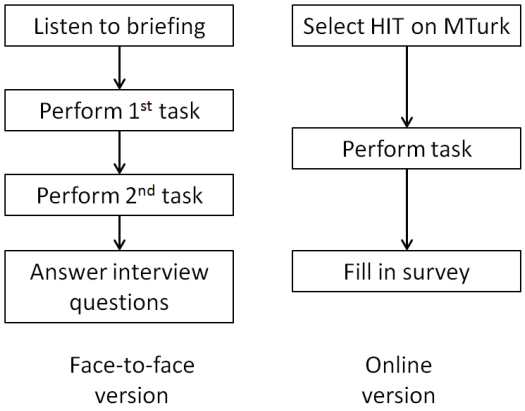
Complex Numbers
Type: *Concept Information*, Difficulty: ★★, Contains: *Definition*
http://www.jgsee.kmutt.ac.th/exell/Numbers/CplxNum.htm
Complex Numbers NUMBER SYSTEMS AND ANALYSIS **... Complex Numbers** are Not Ordered **Complex** Conjugate Powers and Roots **...** Sequences Representation of **Complex Numbers** A **complex number** z has the form **...** cos theta - i.sin theta). **Complex Numbers** are Not Ordered **Complex numbers** cannot be ordered like the **...** a sequence of **complex numbers**. A **complex number** w is a limit **... Numbers ...**

Definition	A complex number z has the form z = x + i.y, where $i^2 = -1$....
	The complex number z may also be written in the form z = r.cos theta + i.r.sin theta , where $r^2 = x^2 + y^2$, cos theta = x/r, and sin theta = y/r....

$a + bi$ ...**A complex number** z has the form **z = x + i.y**, where $i^2 = -1$. ...

Complex Numbers
Type: *Concept Information*, Difficulty: ★★, Contains: *Definition*
http://cnyack.homestead.com/files/MathBgnd/com_nos.htm
Complex Numbers Complex Numbers Cuthbert Nyack Notation A **...** d . If a **complex number** is zero, then both its **...** Polar Form of **Complex Numbers** A **complex number** z = a + jb **...** b/a) Operations on **complex numbers** in polar form **...** roots are very easy if **complex numbers** are in polar **...** 0 to n - 1. **Complex Numbers** in Oscillations and **... ...**

Figure 6.3: Steps in the face-to-face and online versions of the evaluation.



and resources. All these tasks are designed based on our user study and we hypothesize that all of them can be facilitated by (at least) one of the features in our system.

As a baseline, we prepared another system, which is basically our system with the key features hidden. Both systems use the same Lucene backbone to index the same collection of math webpages (our math corpus) and share the same interface except that the information and controls related to the key features are not present in the baseline.

The face-to-face version of the evaluation was conducted in a lab environment in which a desktop computer was provided to the participants. After a short briefing on the goal of our research and the flow of the experiment, we asked the participants to complete one of the search tasks using one of the systems, followed by a second search task using the other search system. For each task, we asked the participants to find five suitable webpages and copy down the URLs of these webpages. In addition, we also asked the participants to copy and paste a fragment of each webpage or write a short comment for it to justify why it is suitable for the task. This additional requirement added to the task realism and provided information to us to decide whether the participants performed the task carefully and correctly.

To ensure fairness, we presented the baseline first to half of the participants and the math system first to the other half. To also simplify the study execution, we always assigned tasks 1 and 3 together and tasks 2 and 4 together.

Table 6.3: Tasks for the math search system evaluation. The descriptions of these tasks as shown in this table were presented to the participants during the evaluation.

Task	Scenario	Description	Related Feature
1	***Search for information about a math concept for learning purposes***	Imagine that you are taking a course in probability. The lecturer told you earlier that the next lesson would be on a math concept called Bayes' theorem. Therefore, you would like to find out its definition, go through a few examples on how it is applied in practice and possibly try to understand how it can be proved.	Resource Categorization: Information Type
2	***Search for information about papers that describe existing research on a math concept***	Imagine that you are a research assistant to a professor in the math department. Today the professor asked you to collect some papers on how matrix diagonalization had been studied and applied in research. Therefore, you need to search for information about papers that describe existing research on matrix diagonalization so that you know what papers to collect next.	Resource Categorization: Resource Type
3	***Search for learning materials for audience without good math background***	Imagine that you have just received a tuition assignment to teach modular arithmetic to a few lower secondary school students. Before meeting them for the lesson, you are planning to send them a few webpages for them to read beforehand. These webpages should be written in a readable way so that the lower secondary school students would be able to get a general idea of what modular arithmetic is about.	Resource Categorization: Readability
4	***Search for expression representations of a math concept***	Imagine that you have just learnt about complex numbers from a math class today and become interested in it. At the end of the class, your teacher told you that there is more than one way to represent complex numbers. Therefore, you would like to find out what the common representations of complex numbers are.	Text-to-Construct Linking

No time limit was imposed on the tasks. On average it took the participants 15 minutes to complete a task. We proceeded to the interview after the participant had completed both tasks.

We have a list of questions (both multiple choice and open-ended questions) for discussion with the participants during interviews. Aside from simple demographics (*e.g.*, their experience in searching for math resources), our questions focus on four topics: 1) how they have performed the tasks, 2) what difficulty they have encountered in performing the tasks, 3) how they have been assisted by the features of the search systems, and 4) how the search systems can be improved. This list of questions can be found in Appendix A.2.

On average the interviews lasted 20 minutes and were not recorded; however, the answers from the participants were compiled during each interview. After the interviews, we consolidated the answers for further analysis. All 81/ participants were rewarded 15 Singapore dollars as a token of appreciation.

As for the online version, we posted the four search tasks as Human Intelligence Tasks (HITs) in Mturk. The HITs contained information about goal of our research, the flow of the experiment, as well as how to access our experiment webpage to perform the evaluation. When interested Mturk workers visited the experiment webpage from a HIT, they would see the detailed description of the corresponding search task and a link to the search system they were supposed to complete the task with. Since the workers were allowed to complete any of the four tasks in any order, we did not explicitly enforce any order in which the search systems were presented to them. Nevertheless, we did make efforts in presenting both systems for each search task as equally often as possible.

For each task, the workers were also asked to find five suitable webpages and copy the URLs of these webpages into a form on the experiment webpage. After completing a task, the workers were asked to fill in a survey which covered the same topics as the interview in the face-to-face version; however, to keep the time needed to complete a HIT within a reasonable limit, the workers were not required to provide justifications for the webpages they had found and the survey only contained selected multiple choice questions from the interview. For more information about the selected questions, please refer to Appendix A.2.

Table 6.4: Numbers of evaluations completed on the math search system and the baseline.

Version	Math Search System				Baseline			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
Face-to-face	11	11	10	10	10	10	11	11
Online	28	26	37	21	33	34	35	25
Total	39	37	47	31	43	44	46	36

For quality assurance, we have implemented some validation and logging mechanisms on the experiment webpage and the search systems to make sure that 1) the participants have performed at least one search on the presented system, 2) all the URLs entered belong to the webpages in our collection and at least one of them should be suitable for the search task (according to our own annotations), and 3) their responses to the questions in the survey match with the search log (*e.g.*, if they claimed to have utilized a particular math feature, one or more entries in the search log should show that the corresponding feature had been activated). The data which failed to meet at least one of these requirements were considered invalid and discarded.

In total, 320 HITs were completed and 81 (25%) of them were discarded. On average, it took the workers 10 minutes to complete a HIT and they were rewarded 0.80 U.S. dollar for each completed HIT.

The protocols for both versions of the evaluation have been reviewed and approved by the Institutional Review Board (IRB) in NUS[6].

6.2.1 Results and Discussions

We have recruited 42 participants for the face-to-face version of the evaluation and 138 (after excluding those who failed to meet the requirements) for the online version. On average, each task was performed on each system 40.4 times. This allows us to perform both qualitative and quantitative analysis on the results. The detailed numbers are as shown in Table 6.4.

In terms of demographics (shown in Table 6.5), the NUS students have a strong background in math. 95% of them know college-level calculus or beyond. 98% of them perform general search daily. 76% of them perform math search a

[6]Reference Code: 12-462E.

Table 6.5: Demographics of the participants.

(a) Math Background	NUS	Mturk
Arithmetic only	0%	11%
High school algebra	0%	12%
High school algebra, trigonometry, some calculus	5%	30%
College calculus	17%	38%
College math beyond calculus	78%	9%

(b) Experience in General Search	NUS	Mturk
A few times per day or more	98%	49%
A few times per week	2%	20%
A few times per month	0%	13%
A few times per year or less	0%	18%

(c) Experience in Math Search	NUS	Mturk
A few times per day or more	7%	2%
A few times per week	21%	8%
A few times per month	48%	25%
A few times per year or less	24%	65%

few times per month or more. In contrast, the math background of the Mturk workers is more diverse and much weaker on average. Only 47% of them know college-level calculus or beyond. They are also much less experienced with general search and math search. Only 49% of them perform general search daily and as much as 65% of them perform math search a few times per year or less.

Despite the difference in background and search experience, many participants have prior experience in performing tasks similar to the ones they did during the evaluation. As shown in Table 6.6, the ratio between the participants who have performed task 1 (*i.e.*, to search for information about a math concept for learning purposes) and those who have not is 2.9:1, indicating that this is a very common task. In comparison, the ratio for task 2 (*i.e.*, to search for information about papers that describe existing research on a math concept) is much smaller (1.1:1) and the lowest among all; however, there are still about half of the participants of this task who have prior experience in it. As for task 3 (*i.e.*, to search for learning materials for audience without good math background), and task 4 (*i.e.*, to search for expression representations of a math concept), the

Table 6.6: Participants' experience in completing tasks similar to the ones in the evaluation.

	Task 1		Task 2		Task 3		Task 4	
	Yes	No	Yes	No	Yes	No	Yes	No
NUS	30	9	18	19	27	20	16	15
Mturk	31	12	24	20	29	17	30	6
Total	61	21	42	39	56	37	46	21
(Ratio)	2.9:1		1.1:1		1.5:1		2.2:1	

ratios are close to 1.5:1 and 2.2:1, both of which indicate that at least 60% of the participants have performed similar tasks before. Although these ratios may be biased (towards the high side) due to the possibility that some Mturk workers may choose to complete only the tasks they have prior experience in, we believe these ratios do verify that our tasks are common math search tasks.

To compare the two search systems quantitatively, in both versions of the evaluation, we required the participants to give the following two ratings:

- The effectiveness of the search engine for completing the given task (on a 5-point scale with 1 being very ineffective and 5 being very effective)
- The perceived difficulty in completing the task using the given search engine (on a 5-point scale with 1 being very easy and 5 being very difficult)

Between these two ratings, effectiveness is more important since, as mentioned at the beginning of this section, the objective of this evaluation is to determine whether the key features are effective in helping math seekers in their searches. As such, we apply two-tailed, unpaired Student's t-test with $p < 0.01$ on the effectiveness ratings to determine the statistical significance of the differences between the math search system and the baseline. The results are as shown in Table 6.7 and Table 6.8.

The overall effectiveness rating for the baseline is 3.46, indicating that its effectiveness is above normal (*i.e.*, 3). It achieves the highest effectiveness rating on task 1 (3.95). In our opinion, this shows that the baseline performs reasonably well in helping the participants find the information and resources about a math concept; however, considering the fact many participants have prior experience in this task and the perceived difficulty is the lowest (2.23) among all tasks, we believe that this relatively high effectiveness rating may be partly due to familiar-

Table 6.7: Average effectiveness ratings of the math search system and the baseline. The ratings are given on a 5-point scale, with 1 being very ineffective and 5 being very effective. M and B in the table stand for math search system and baseline respectively. Bolded pairs of ratings for a particular task and participant group indicate that the difference between the two ratings are statistically significant ($p < 0.01$).

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	4.09	3.80	4.00	3.50	**4.50**	**3.09**	4.10	3.21	**4.17**	**3.21**
Mturk	4.11	4.00	**4.19**	**3.29**	4.00	3.62	**4.29**	**3.24**	**4.15**	**3.54**
Combined	4.10	3.95	**4.14**	**3.34**	**4.11**	**3.35**	**4.23**	**3.20**	**4.32**	**3.46**

Table 6.8: Average perceived difficulty ratings of the math search system and the baseline. The ratings are given on a 5-point scale, with 1 being very easy and 5 being very difficult. M and B in the table stand for math search system and baseline, respectively.

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	2.00	2.20	2.82	2.50	2.50	3.45	2.17	2.81	2.37	2.74
Mturk	2.29	2.24	2.65	3.09	2.22	2.63	2.43	2.68	2.40	2.66
Combined	2.21	2.23	2.70	2.95	2.28	2.82	2.37	2.72	2.39	2.68

ity with the task. As can be observed in task 2 and 4, which are two less familiar tasks for both groups, the effectiveness ratings are also lower. In addition, the two groups disagree on its effectiveness on task 3: NUS students find it normal (3.09) while the Mturk workers still find its effectiveness above normal (3.62). This disagreement can also be observed in the perceived difficulty ratings (3.45 for NUS students and 2.63 for Mturk workers). During our interviews with the NUS students, we discovered that they found task 2 difficult because they were having difficulty in thinking in the shoes of the lower secondary school students to decide which webpages might be suitable. This was much less of a problem for the Mturk workers because many of them had similar math background with lower secondary school students. Nevertheless, the fact that the Mturk workers have weaker background may also be the main reason why they find task 2 (*i.e.*, finding research papers) most difficult.

In contrast, the math search system achieves 4.10 or above on effectiveness ratings for all the tasks when we combined the two groups together. In other words, it is all rounded and often rated higher than effective. When compared to the baseline, its effectiveness ratings are consistently higher on all tasks. The

Table 6.9: Average accuracy scores of the answers given by the participants. The scores are awarded based on how many out of the 5 webpages found by the participants are indeed suitable. Each unsuitable/partially suitable/suitable webpage is worth 0/0.5/1 mark. M and B in the table stand for math search system and baseline respectively.

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	4.82	4.75	5.00	4.70	4.54	4.32	4.81	4.59	4.79	4.59
Mturk	4.67	4.52	3.03	1.91	4.01	3.81	4.02	3.82	3.94	3.52
Combined	4.72	4.57	3.62	2.54	4.12	3.93	4.28	4.06	4.19	3.78

differences between the two systems on effectiveness ratings are statistically significant except for task 1 which the baseline already performs quite well for. The math search system also leads to lower perceived difficulty ratings. We believe these results are good validation that the math search engine better facilitates math search than the baseline.

Since both effectiveness and perceived difficulty ratings are subjective measures, we have also evaluated both systems based on one additional objective measure: accuracy. To judge accuracy, we manually checked through all the answers given by the participants to assess whether they are unsuitable, partially suitable or suitable. Each unsuitable/partially suitable/suitable webpage contributes 0/0.5/1 mark to the accuracy score of the answers.

As shown in Table 6.9, the average accuracy scores of the math search system are consistently higher than those of the baseline. In other words, the math search engine indeed helped the participants to find more suitable webpages than the baseline in all tasks.

In addition, by analyzing the answers from the interviews and surveys, we have noted that a number of the participants did not notice the key features at all when they performed a task using the math search system. When we probed the NUS students for the reason during the interviews, some of them replied that they were too used to reading the result titles, URLs and snippets that they simply ignored anything else, while some others replied that they liked to quickly move to read the contents of the webpages instead of spending time with the search system. Therefore, we have noted that more work has to be done on the interface so that the key features can optimally gain the user's attention.

Table 6.10: Numbers of participants who did not notice the key features in the math search system.

	Task 1	Task 2	Task 3	Task 4
NUS	1 (9%)	1 (9%)	1 (10%)	3 (30%)
Mturk	5 (18%)	4 (15%)	7 (19%)	5 (24%)
Total	6	5	8	8

Table 6.11: Adjusted numbers of evaluations completed on the math search system and the baseline.

Version	Math Search System				Baseline			
	Task 1	Task 2	Task 3	Task 4	Task 1	Task 2	Task 3	Task 4
Face-to-face	10	10	9	7	11	11	12	14
Online	23	22	30	16	38	38	42	30
Total	33	32	39	23	49	49	54	44

As these participants did not notice the key features at all and both systems return the same results by default, we present the adjusted evaluation results which factor out such participants. The adjusted numbers of evaluation completed, effectiveness ratings, perceived difficulty ratings and accuracy scores are shown in Table 6.11 to 6.14. The same statistical significant test is also applied on the adjusted effectiveness ratings.

After the adjustments, the gap between the math search system and the baseline widens on all metrics. All the differences in effectiveness ratings (when the two groups are combined) are now statistically significant. The gaps in effectiveness ratings for task 2, 3 and 4 are 0.9, 1.0 and 1.16. In other words, on these tasks, the math search system is around one level more effective compared to the baseline. These results are clear indications that when the key features are

Table 6.12: Adjusted average effectiveness ratings of the math search system and the baseline. The ratings are given on a 5-point scale, with 1 being very ineffective and 5 being very effective. M and B stand for math search system and baseline respectively. The participants who used the math search system but failed to notice the key features are considered to have used the baseline instead. Bolded pairs of ratings for a particular task and participant group indicate that the difference between the two ratings are statistically significant ($p < 0.01$)

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	4.10	3.82	4.00	3.45	**4.50**	**2.50**	4.10	3.21	**4.17**	**3.25**
Mturk	**4.35**	**3.87**	**4.32**	**3.32**	**4.20**	**3.58**	**4.50**	**3.30**	**4.34**	**3.51**
Combined	**4.27**	**3.86**	**4.25**	**3.35**	**4.31**	**3.31**	**4.43**	**3.27**	**4.32**	**3.45**

Table 6.13: Adjusted average perceived difficulty ratings of the math search system and the baseline. The ratings are given on a 5-point scale, with 1 being very easy and 5 being very difficult. M and B stand for math search system and baseline respectively. The participants who used the math search system but failed to notice the key features are considered to have used the baseline instead.

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	2.00	2.18	2.70	2.64	2.56	3.33	2.43	3.07	2.42	2.81
Mturk	2.26	2.26	2.57	3.03	2.03	2.69	2.44	2.63	2.32	2.65
Combined	2.18	2.24	2.61	2.94	2.15	2.83	2.43	2.77	2.34	2.70

Table 6.14: Adjusted average accuracy scores of the answers given by the participants. The scores are awarded based on how many out of the 5 webpages found by the participants are indeed suitable. Each unsuitable/partially suitable/suitable webpage contributes 0/0.5/1 mark to the accuracy score of the webpages found. M and B stand for math search system and baseline respectively. The participants who used the math search system but failed to notice the key features are considered to have used the baseline instead.

	Task 1		Task 2		Task 3		Task 4		Overall	
	M	B	M	B	M	B	M	B	M	B
NUS	4.80	4.77	5	4.73	4.61	4.21	4.86	4.64	4.81	4.59
Mturk	4.65	4.55	3.45	1.79	4.20	3.71	4.19	3.76	4.12	3.46
Combined	4.70	4.60	3.94	2.45	4.29	3.82	4.39	3.89	4.33	3.69

noticed (and possibly made use of), they contribute greatly to the effectiveness of the search system.

To perform a more detailed analysis on the key features, we divide them into four sub features: Information Type Categorization (ITC), Resource Type Categorization (RTC), Readability Categorization (RC) and Text-to-Expression Linking (T2E). We further distinguish two types of implementation (passive/active) for each sub feature in the math search system as shown in Table 6.15.

In passive implementations, the categorization and linking outputs are dis-

Table 6.15: Types of implementations of sub features

Sub feature	Passive Implementations	Active Implementations
Information Type Categorization	Display of information type in the search results **(ITCp)**	Filters for information type **(ITCa)**
Resource Type Categorization	Display of resource type in the search results **(RTCp)**	Filters for resource type **(RTCa)**
Readability Categorization	Display of readability in the search results **(RCp)**	Filters and sort options for readability **(RCa)**
Text-to-Expression Linking	Display of linked expressions the search results **(T2Ep)**	Searching with linked expressions **(T2Ea)**

Table 6.16: Numbers of participants noticing and utilizing the sub features and their effective ratings. The acronyms (*e.g.*, ITCp and ITCa) refer to the type of implementation of the sub features as listed in Table 6.15.

	ITCp	ITCa	RTCp	RTCa	RCp	RCa	T2Ep	T2Ea
NUS								
Noticed	10	9	7	9	9	8	7	6
Utilized	7	3	6	5	7	6	7	6
Rating	4.14	5	4.17	4.67	4.71	4.83	4.57	5
(Micro-average)	4.40		4.39		4.77		4.77	
Mturk								
Noticed	22	16	22	20	29	26	13	14
Utilized	22	13	22	16	29	16	13	11
Rating	4.41	4.46	4.50	4.25	4.41	4.63	4.46	4.55
(Micro-average)	4.43		4.39		4.49		4.50	
Combined								
Noticed	32	25	29	29	38	34	20	20
Utilized	29	16	28	21	36	22	20	17
Rating	4.34	4.56	4.42	4.35	4.47	4.68	4.50	4.71
(Micro-average)	4.42		4.39		4.55		4.59	

played directly in the search results; no extra effort (besides reading them) is required from users. In contrast, in active implementations, the categorization outputs are for filtering and sorting, while the linking outputs are for searching; additional efforts in activating these features are required. In both versions of the evaluation, we required the participants to answer whether they have noticed and utilized the sub features and rate these features on their effectiveness. The consolidated statistics are as shown in Table 6.16.

In general, all passive implementations are well-utilized: More than 90% of the participants who noticed these implementations made use of them in completing the search tasks. According to our interviews with the NUS students, almost all of them who noticed the additional information confirmed that it helped them decide which webpages to read in more detail (task 1-3) and increased their knowledge about the math concept they were searching for (task 4). Among those who noticed but did not utilize the implementations, the two common reasons are: 1) They found the result title and snippet more important and paid a lot more attention to them, and 2) they noticed and applied the active implementations first (*e.g.*, to filter out non-papers results in task 2 and less readable results in task 3). Given the fact that all NUS students are heavy users

of general search engines which do not display additional information, we believe the users of the math search system would be more aware of the additional information if they were more familiar with the system.

In contrast, the percentages of participants who made use of the active implementations after noticing them are much lower, ranging from 64% (task 1) to 85% (task 4). Through our discussions with the participants, we observed that, for those participants who do not commonly use active implementations, they would consider using those implementations only when there are too many (irrelevant) results even after some low-cost alternatives (*e.g.*, adding keywords) have been attempted. Therefore, we believe one way to increase the utilization of active implementations is to lower their cost. For example, the system may automatically enable some of the filters based on the query keywords (*e.g.*, filter out non-paper results when the keyword 'paper' is used in the query).

Nevertheless, the overall effectiveness ratings for all the sub features are 4.39 or above[7], indicating that they are effective for completing the tasks.

6.2.2 Future Work

To sum up, our evaluation on the math search system shows that it outperforms the baseline significantly in terms of effectiveness. It also helps to lower the perceived difficulty of the common math search tasks and allow users to find more suitable webpages. We have even received an appreciation email (Appendix A.3) from a Mturk worker expressing his interest in using the system to find math materials for his children (a scenario very similar to task 3 in the evaluation).

All math features are found to be effective for completed the tasks, despite the fact that some of them are less utilized due to habitual behaviors and the additional efforts required for utilization.

As such, an immediate direction for future research is to work on improving the visibility of the math features and lower the cost of using them. For this purpose, automated detection of what math features can be applied for a particular search based on queries and user behaviors would be important. With such detection, the system may then prompt users on the suitable math features or

[7]See the bottom row of Table 6.16.

trigger those features implicitly.

In addition, the current size of our math webpage collection is still too small for public use. In future, we plan to expand the collection so that the system would be more ready for further research, evaluation and deployment.

6.3 eEvidence System for Evidence-based Practice in Healthcare

Recall from Section 3.1 that Evidence-based Practice (EBP) is defined as meeting the information needs of practitioners with the synthesis and critical appraisal of applicable and valid literature. The application of EBP can be divided into two stages: 1) evidence gathering and selection, and 2) practice implementation and outcome evaluation. The first stage is crucial because the quality of the evidence gathered directly influences the downstream best practices.

To ensure proper gathering and selection of evidence in healthcare, most EBP literature suggests an active search process that includes the formulation of clinical questions, the search for evidence and the appraisal of evidence [Fineout-Overholt et al., 2005; Brady and Lewin, 2007]: A clinical question is first formulated using PICO elements [Melnyk and Fineout-Overholt, 2000] (*i.e.*, patient, intervention, comparison and outcome). With this question in mind, keyword searches on EBP collections [Oremann, 2007] (such as CINAHL, Medline and Embase) are conducted to locate candidate articles. Lastly, the candidate articles are appraised critically, on criteria such as applicability and validity.

Despite the fact that such a proactive process is useful, it is often too difficult and time-consuming for the healthcare practitioners. This is due to two reasons:

First, EBP collections exist largely in isolation (*i.e.*, are not connected with each other and only searchable via their own interface) and commonly require subscription. As a result, healthcare practitioners would not only need to perform searches in these resources one by one, but also initiate separate searches for the articles which appear to be relevant but are not accessible from the current collection. Several linking schemes have been developed to alleviate this

difficulty. For example, Digital Object Identifiers (DOIs)[8] make it possible to quickly locate a digital copy of an article in a particular EBP collection; however, without taking into account the location or the affiliation of the practitioners, they often fail to resolve to a copy that the practitioners have access to (*i.e.*, commonly known as the appropriate copy problem) [Beit-Arie et al., 2001]. More recently, the OpenURL framework [Apps and MacIntyre, 2006] addresses this limitation by building knowledge bases that contain the availability and accessibility information of articles. Nevertheless, the main caveat of this scheme is that it requires much effort to ensure that the metadata encoded in OpenURLs are accurate and consistent [Chandler et al., 2011], and that the knowledge bases are accurate, comprehensive and up-to-date [Wikipedia, 2012a]. Therefore, we believe this difficulty in dealing with multiple isolated EBP collections with subscription barriers is going to remain until a cost-effective solution is found.

Second, current search engines are limited in their capabilities for evidence gathering and selection. While publicly-accessible generic search engines can search different resource collections and find free materials, their search results are hard to navigate through unless proper categorization is done to group the resources by type. In contrast, specialized search engines are designed specifically for medical search with comprehensive medical knowledge and metadata but are often restricted in accessibility and meant to be used exclusively for one particular healthcare EBP collection.

These difficulties are coupled with the fact that most healthcare practitioners have to spend much of their time taking care of patients [Bond, 2005] and may not be well-trained in searching. As a result, they are often unable to follow a routine process to stay up-to-date with the literature.

An alternative to active search is to postpone the integration of current research practices until later in the healthcare workflow and delegate the selection process to an automated system. For example, some knowledge-based clinical decision support systems link patient records to medical knowledge in knowledge bases to facilitate downstream decision making [Bakken et al., 2008]. A major problem with such systems is to ensure that the evidence in knowledge bases

[8] http://www.doi.org/doi_handbook/

always incorporates current research findings [Sim et al., 2001]. More recently, meta-search systems such as InfoBot [Demner-Fushman et al., 2008] present information retrieved from five healthcare EBP collections based on the biomedical terms extracted from the patient records. While these systems save the practitioners' trouble of searching through these resources individually, they lack the flexibility to allow the practitioners to customize the search process or explore evidence from other collections.

In summary, for evidence gathering and selection, active search is the recommended practice, but the isolation of EBP collections, as well as the choice between generic and specialized search engine, often complicates the process and makes it less desirable in the interest of time. In contrast, delegating the process to automated systems helps to save time but such systems are challenging to build and often lacking in flexibility and coverage.

Our search system for healthcare aims to address these limitations. It allows healthcare practitioners to curate their own sets of relevant articles for EBP, under an organized framework. In this section we discuss our framework's three key novel features:

Feature 1: Harvesting EBP articles by periodic crawling. Our system crawls freely accessible EBP articles from the Web to create an EBP collection from which practitioners can curate materials. This crawling ensures that our collection covers a variety of EBP articles and contains the latest research findings.

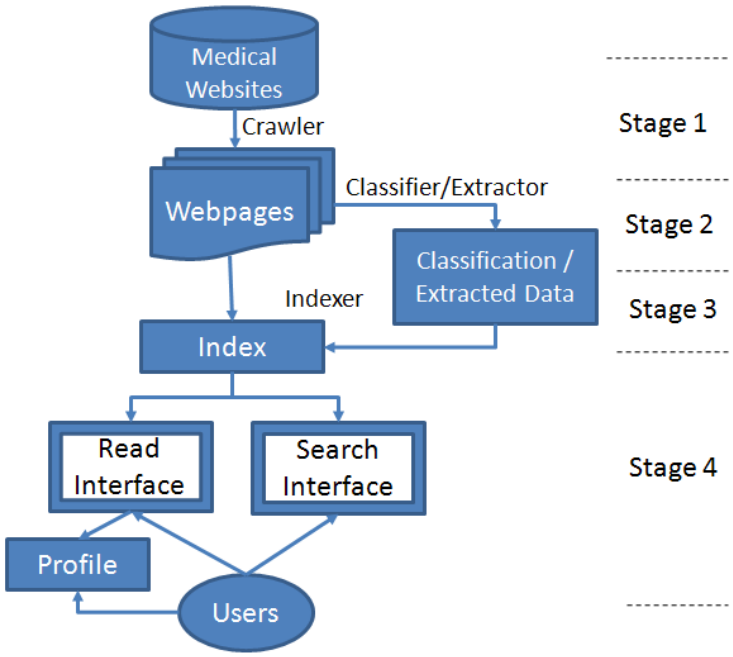
Feature 2: Automated article classification and key information extraction. Automated classification of articles helps to filter out irrelevant documents and separate the rest into different categories, assisting practitioners to zoom to relevant articles quickly. Key information is additionally extracted to assist practitioners in assessing the applicability and validity of candidate articles.

Feature 3: Dual active/passive user interface. Our system presents two interfaces to cater for both active and passive search. It allows practitioners to choose their preferred interaction mode based on their goal and time available.

6.3.1 System Description

We will first describe the architecture of our system and then explain the three key features in detail.

Figure 6.4: Architecture of the eEvidence system.



The architecture of our system (Figure 6.4) consists of four stages:

Stage 1 We use the Nutch crawler[9] to conduct periodic crawls on manually selected EBP collections to obtain a collection of EBP articles.

Stage 2 We then apply machine-learned classifiers and extractors to determine the resources' types and extract various information (*e.g.*, patient demographics, study design, and year of publication) from them.

Stage 3 We again employ Lucene to index the resources together with the results from classification and extraction.

Stage 4 Users access the indexed information through either the search or read interfaces, depending on their information seeking modes.

Feature 1: Harvesting EBP articles by periodic crawling. To construct the resource collection for our system, we asked several healthcare practitioners

[9] http://lucene.apache.org/nutch/

from National University Hospital to select a set of authoritative websites as starting points for crawling. Among the 94 selected websites, our system then crawls the contained webpages from the ones that permit crawling. This crawling can be repeated periodically to ensure that the latest documents are ingested.

There are two reasons why we choose to construct our resource collection by crawling: First, this method works with all web-accessible materials. Therefore, our system has no problem harvesting resources of different types or from different websites. This ensures comprehensive coverage of our resource collection. Second, periodical crawling addresses the freshness problem, alleviating the problem of keeping the indexed collection up-to-date.

Feature 2: Automated article classification and key information extraction. While crawling collects webpages from curated sites, not all pages of a site are relevant, primary research. Irrelevant pages, such as tables of contents and help pages, need to be filtered. Our system automatically accomplishes this. In addition, for primary research articles, the system further subcategorizes them as abstract-only or full-text articles. This information is propagated into the downstream user interface, allowing users to choose which type to view.

Therefore, we apply supervised learning techniques to classify the webpages into three categories: the abstract of a research article, the full text of a research article and any other webpages (to be discarded).

To build the classifiers, we have randomly chosen and annotated 500 webpages from the harvested resources as our training data. The features extracted are similar to the ones used for the resource type categorization of math webpages. CRF is again used as the learning methodology. This classifier is applied onto the rest of the webpages to determine their types.

Moreover, through our discussion with practitioners, we have noticed that the following information about the webpages also plays a part in the evidence selection process. Therefore, after classification, we extract such information from the webpages themselves and the crawl data:

Key sentences and keywords: These key sentences and keywords contain and represent information pertinent to EBP. They allow users to judge

the applicability and validity of the articles without having to read them in full. The extraction of these key sentences and keywords is done as described in Chapter 3.

Year of publication: Newer publications are preferred, as they present latest findings. This is extracted from the webpages using regular expressions.

Time added: The system tracks when the resources are added so that it can inform users about newly added resources since their last login. This information is obtained directly from crawl data.

URL: Besides serving as a link to the original resource, it also gives the provenance of the resource, which has been shown to be useful in judging its trustworthiness. This information is obtained directly from crawl data.

Feature 3: Dual active/passive user interface. Our system keeps users updated with a passive read interface, which recommends relevant articles to them periodically, based on their interests saved in a stored profile. A separate, active searching interface allows them to pose queries to retrieve relevant articles. The two modes are interlinked to allow seamless change of interaction modes.

Passive Read Interface: To make use of the read interface, healthcare practitioners need to construct their user profiles. They key in their interests using primary and secondary keywords. The primary keywords represent the topics of interest (usually names of symptoms or diseases), while the secondary keywords represent the relevant aspects of the topics. For example, a healthcare practitioner who is interested in how cancer has affected the quality of life of patients may put "cancer" as the primary keyword and "quality of life" as the secondary keyword.

With their interests encoded into profiles, our system automatically presents the latest relevant articles whenever they access the system via the read interface. Figure 6.5 shows how the read interface highlights recent results that have been added to the system since their last login. Filtering is also enabled. If they are only interested in a particular type of articles or the

ones published within a particular period of time, they can employ filters to customize the results dynamically.

Aside from standard search engine snippet metadata, our system also shows the pertinent extracted information – key sentences and keywords, year of publication, article type and time added as shown in Figure 6.5 and 6.6 – assisting users in selecting suitable articles.

Active Search Interface: This interface (Figure 6.7) caters for users who are actively searching and is designed with similar conventions to generic search engines, but with enhanced support for query formulation.

Similar to the profile keywords in the read interface, a query in the search interface is a combination of primary keywords (used to search for articles relevant to a certain topic) and secondary keywords (used to filter out articles that are irrelevant to the desired aspects of the topic). More complex queries can be constructed by joining multiple subqueries with Boolean operators. Users may also specify additional constraints (*e.g.*, published in recent 5 years) for the queries using filters.

All query-related information, such as keywords used, filters applied, time of search and number of results returned are saved in the search history to assist users in keeping track of the searches they have conducted.

6.3.2 Evaluation and Future Work

Due to the specialist nature of healthcare practitioners and their busy schedules, it is difficult to recruit a sufficient number of them for quantitative evaluations on our system. Therefore, we have chosen to engage those practitioners who have assisted us in collecting EBP resources in qualitative evaluations instead. As a start, we have asked them to informally evaluate our system by using it to search for the EBP articles they are interested in.

The comments we have received from them are mostly positive and encouraging. In particular, the classification and extraction feature was most appreciated by them, as it allowed them to focus exclusively on the full text articles and see the information relevant to EBP directly. This is a good verification that

Figure 6.5: Read interface of the eEvidence system showing latest articles on ventilator-associated pneumonia.

Figure 6.6: Display of extraction results in the eEvidence system to assist users in applicability and validity assessment. The sentence in this figure is extracted from the first result in Figure 6.5 and can be viewed through the "key text" hyperlink at the end of the result. The types of information it contains are shown on its left while the extracted keywords in it are highlighted based on their types following the color scheme on its right. (Note that not all types of keywords are present in this sentence.) The types of key sentences and keywords extracted are described in detail in Chapter 3.

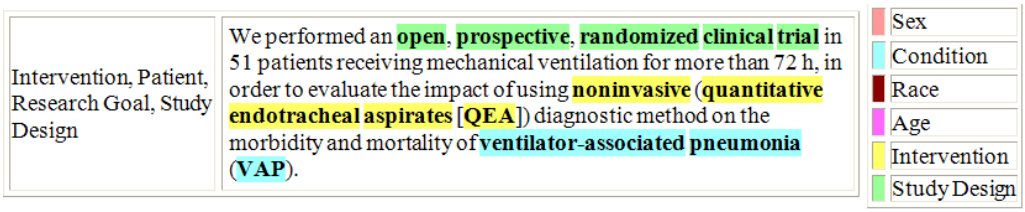
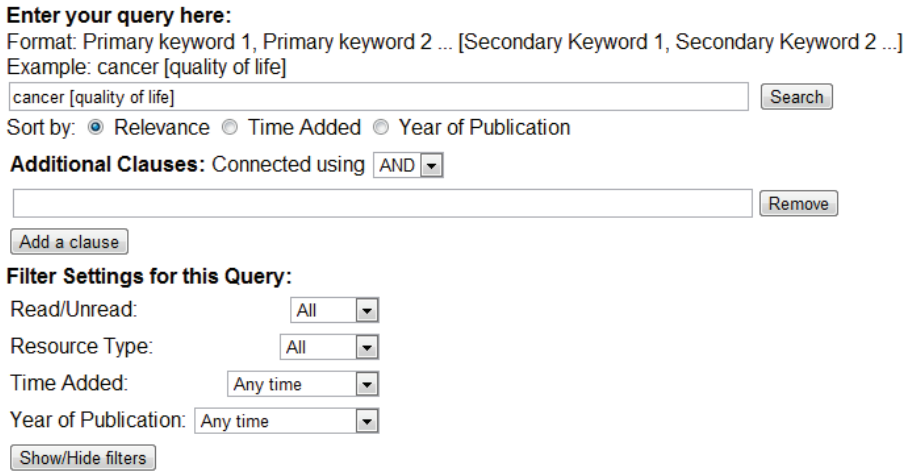


Figure 6.7: Query formulation tool in the search interface of the eEvidence system.



Resource Categorization at multiple granularities is beneficial to domain-specific searchers. In addition, the functions specifically designed to support their workflow had also attracted their attention. For example, they expressed interest in the search history function in the search interface because the recorded information would come in handy for writing the search methodology section in their systematic review. Lastly, they also commented that they were able to find free full text articles which were not found in other medical databases. This is a positive indication that harvesting EBP articles with crawling can be advantageous.

While these results are indicative rather than informative and comprehensive, we believe they do suggest the usefulness of our system.

The main challenge we are facing now is that the amount of articles in our collection is still small compared to existing EBP collections. Currently we only have the corpus for key information extraction, which consists of 19,893 medical abstracts and full text articles, in our collection. Many other websites have been recommended by the healthcare practitioners but most of them are not crawlable due to their robot exclusion policy.

With the current collection, the healthcare practitioners noted that a more thorough evaluation of the system is possible only if more documents can be indexed so that they could accomplish a realistic, sizeable task – such as a literature review on a concrete topic – with our system. As such, we plan to work with NUS libraries to obtain more healthcare materials from the medical databases they have subscribed to. With more materials in our collection, we can then proceed to perform a full-fledged user evaluation to get a better idea of whether our features, especially the extraction of key information, are indeed helpful in facilitating the implementation of EBP.

6.4 Discussion

As demonstrated by the two domain-specific search systems we have built, the findings from our research can be applied to support the information seeking behaviors of domain-specific searchers. However, to perform full-fledged evaluations on these systems and bring them to production, dataset expansion is one of the most important directions for future work.

Chapter 7

Conclusion

Keyword search is ineffective in locating domain-specific resources. Our user study has discovered two key issues pertinent to it in domain-specific IR. First, different modes of domain-specific resources are not recognized, leading to disorganized, hard-to-navigate search results in response to keyword searches. Second, while it is desirable to make domain-specific constructs searchable and relevant in ranking, users still prefer to use text keywords over other input modalities.

To improve domain-specific IR in general without expensive domain knowledge sources, problems related to these issues need to be identified, examined and then addressed in a generic manner. Moreover, the resulting findings need to be translated into features for domain-specific search systems. Therefore, our research has the following three specific goals as introduced in Chapter 1.

1. To identify prominent problems in domain-specific IR. These problems should be sufficiently common yet addressing them should facilitate domain-specific IR.

2. To address the identified problems in a generic manner so that different instances of such problems in different domains can be addressed similarly.

3. To incorporate the research findings into domain-specific search systems. This helps to verify the usefulness of our research and improve domain-specific IR in practice.

In the following sections, we will recap the contributions we have made towards these goals and then discuss about directions for future research.

7.1 Contributions

Identifying Two Prominent Problems in Domain-specific IR. To address the two issues with keyword search, we have identified two research problems that are prominent in many domains:

- **Resource Categorization** refers to the problem of categorizing domain-specific resources with respect to different facets at both coarse-grained and fine-grained levels. Due to the explosive growth of domain-specific resources, in almost any domain, there is a need to at least categorize the resources at a coarse-grained level (*e.g.*, resource-level) based on the intent (*e.g.*, learning-oriented vs. research-oriented) and the background (*e.g.*, novice vs. expert) of the target audience. To cater for more specific needs, such as applicability and validity assessment, more fine-grained categorizations (*e.g.*, at sentence-level and word-level) may be required as well. Proper handling of this problem means that a search engine can better meet specific user needs by directing task-relevant resources to users and organize search results by more pertinent metadata criteria.

- **Text-to-Construct Linking** refers to the problem of resolving text keywords to their relevant domain-specific constructs. This problem is prevalent in many domains where domain-specific constructs exist. Common examples of domain-specific constructs include expressions (math), molecular structures (chemistry) and DNA (biology). Proper handling of this problem makes domain-specific constructs searchable by text keywords, which in turn, can enable constructs to properly influence relevance ranking in search results, without troubling users to input them in potentially awkward construct syntax.

Providing Domain-independent Approaches to Address the Two Prominent Problems. We have observed correlations among various characteristics of domain-specific resources and captured such information in a multi-layered graph as shown in Chapter 1. Following this graph, we examine the problems of Resource Categorization and Text-to-Construct Linking, and seek for domain-

independent approaches to address them.

For Resource Categorization, we use the key information extraction problem for evidence-based practice in healthcare as a case study on the categorization of correlated nominal facets. We have compared four different models for exploiting the correlation to inform the classification process. The joint inference model works well in allowing two classification tasks to benefit from each other at the same time; however, data filtering needs to be applied to alleviate its computational cost and reduce the noise in training data. This utilization of classification results at two different levels to inform each other is generically applicable in any domain with correlated nominal facets to be categorized.

On the other hand, we use the readability measurement problem for domain-specific resources as a case study on the categorization of ordinal facets. By correlating the readability of domain-specific resources with the difficulty of domain-specific concepts, we are able to use an iterative computation algorithm to estimate their values from each other. This approach performs well even when compared to supervised learning approaches, and can be ported to other domains easily since it does not rely on expensive domain knowledge sources or even an annotated corpus.

Modeling Text-to-Construct Linking is more complicated, as it requires to link domain-specific concepts to relevant constructs in domain-specific resources, and then select the better ones for display and retrieval. Through a corpus study in math, we have identified a set of possible relations between domain-specific (*i.e.*, math) concepts and constructs (*i.e.*, expressions) and collected statistics that help to characterize the nature of the linking problem. We link concepts to their representations in constructs through supervised classification and then rank the linked constructs by construct-TF.IDF. Our results show that satisfactory linking performance can be achieved with non-domain-specific features, while construct-TF.IDF works well in selecting more specific and informative constructs for display and retrieval. Since our approach for this problem does not rely on expensive domain knowledge sources, it is also domain-independent and can be adapted to perform Text-to-Construct Linking in other domains.

Implementing Two Domain-specific Search Systems. To demonstrate the

applicability and usefulness of our research, we have implemented two domain-specific search systems, one for math and the other for healthcare, based on the findings from our research on Resource Categorization and Text-to-Construct Linking. The math system incorporates the categorizations of resource type, information type and readability, which allow for better organization of search results. It also links math concepts to their expression representations, which alleviates the need for expression input yet maintains the use of expressions for display and retrieval. The healthcare system categorizes resources at multiple granularities to extract key information for applicability and validity assessment in evidence-based practice. In addition, it is equipped with features (*e.g.*, dual interface) for better workflow integration.

Our evaluation on the math system shows that it is significant more effective than a general search baseline on four common math search tasks. It lowers the perceived difficulty of the tasks and allows users to find more suitable webpages. In addition, all the math features in the system have been rated as effective or above. As for the healthcare system, it has received mostly positive and encouraging comments during the informal evaluation; however, more documents need to be indexed to allow for a full-fledged evaluation.

Both systems can serve as platforms for domain-specific IR research and be expanded into practical systems for public use in future.

7.2 Future Work

Besides the possible ways to improve Resource Categorization and Text-to-Construct Linking as listed in the respective chapters, we believe two major directions for future research in domain-specific IR are as follows:

User-centric Development. As mentioned in Section 2.1, domain-specific searchers have specialized needs because they have different roles and exhibit a wide spectrum of domain knowledge. Without taking their needs into consideration, domain-specific IR research may head into directions that are not immediately useful in facilitating domain-specific search. Moreover, domain-specific searchers are the ultimate judge of whether the problems they have encountered

are addressed by the proposed approaches. Therefore, they need to be involved at both the problem formulation and evaluation stages of domain-specific IR research. Our user study in math and our experience in working with healthcare practitioners have served as a good starting point for involving domain-specific searchers into our research process. In future, we plan to maintain long-term relationships with a pool of domain-specific searchers of diverse roles and backgrounds so that we can understand and address their needs better.

Cross-domain Investigation. Our research in math and healthcare allows us to identify two prominent problems and propose domain-independent approaches for these problems. However, given the fact that many other domains exist and each of them is special in its own way, it is likely that there are more variations of the problems we have examined and other prominent problems to be tackled. Therefore, it is important to continue this process with more domains so that domain-specific IR can be improved in general instead of only in a few domains. To this end, we hope to carry out cross-domain user studies and comparative studies of domain-specific search systems in future to identify more common problems in domain-specific IR. With such problems identified, we can then work on finding domain-independent approaches for them by examining concrete instances of these problems from several different domains all at once. Last but not least, we would like to perform cross-domain evaluations to determine the effectiveness and domain independence of the proposed approaches, as well as to ascertain the need for specialization in dealing with a particular instance of these problems in a specific domain.

Appendices

A.1 Examples of Nodes and Edges in the Correlation Graph

Please see the next few pages for the tables of examples.

Examples of nodes in the concept layer.

Name	Definition	Node Type	Value Type	Examples
Concept Type	The nature of a concept.	Hidden	Nominal	Math: areas (geometry and number theory), operations (Fourier transform) and theorems (Pythagorean theorem). Medicine: diseases (diabetes), injuries (bruise), substances (vitamin) and symptoms (snoring).
Difficulty	The amount of prerequisite knowledge required to understand a concept.	Hidden	Ordinal	More difficult to less difficult: integration and differentiation > exponentiation > addition and subtraction.
Genericity	The amount of domain knowledge referred to by a concept.	Hidden	Ordinal	More generic to less generic: geometry > differential geometry > Riemannian geometry.
Relatedness	The amount of connections a concept has with the other concepts in the same resource.	Hidden	Ordinal	More related to less related: Pythagorean theorem discussed with geometric concepts such as right triangles and hypotenuse > General discussion on theorems in different areas with Pythagorean theorem cited as an example.

Example of nodes in the resource layer.

Name	Definition	Node Type	Value Type	Examples
Resource Type	The genre of a resource defined based on the types of information it contains and how such information is organized.	Hidden	Nominal	Tutorial, encyclopedia, discussion, paper, course website, resource hub.
Readability	The amount of prerequisite knowledge required to understand a resource.	Hidden	Ordinal	More readable to less readable: an explanation of modular arithmetic as clock arithmetic > a discussion of modular arithmetic in the context of ring theory.
Specificity	The level of details at which the concepts are discussed in a resource.	Hidden	Ordinal	More specific to less specific: a tutorial on matrix diagonalization with a detailed definition, examples and exercises > a dictionary entry on matrix diagonalization with only a concise definition.
Cohesion	How well the ideas in a resource connect to each other.	Hidden	Ordinal	More cohesive to less cohesive: a discussion on several closely-related concepts > a description of several independent sub-topics in an area.
Resource Word Sequence	Whether a particular sequence of words appears in a resource.	Observable	Nominal	-
Average Sentence Length	The average number of words per sentence in a resource.	Observable	Ordinal	-

Examples of edges in the resource layer.

Pair	Explanation
Resource Type & Observable Characteristics	Resource-level observable characteristics, such as word sequence, may serve as indicators for resource type.
Resource Type & Readability	The type of a resource is indicative of its readability. For example, a tutorial is usually more readable than a paper on the same concept since the former is meant for beginners while the latter for experts.
Resource Type & Specificity	The type of a resource is indicative of its specificity. For example, a paper is usually more specific than an encyclopedia page on the same concept since the former covers only a specific aspect of it while the latter may need to cover many.
Readability & Observable Characteristics	Resource-level observable characteristics, such as average sentence length and average number of syllables in the words, may serve as indicators for readability. [Flesch, 1948]
Specificity & Observable Characteristics	Resource-level observable characteristics, such as total length and average number of sentences in a paragraph, may serve as indicators for specificity.
Cohesion & Observable Characteristics	Resource-level observable characteristics, such as ratio of causal particles to causal verbs, may serve as indicators for cohesion. [Mcnamara et al., 2002]

Examples of nodes in the segment layer.

Name	Definition	Node Type	Value Type	Examples
Segment Type	The types of information a segment contains or represents.	Hidden	Nominal	Math: definition, example and proof. Healthcare: patient demographics, study design, intervention.
Relation Type	The type of semantic relation that exists between two segments.	Hidden	Nominal	Math: representation (*i.e.*, a math concept and its representation in expression) and property (*i.e.*, a math expression and a concept that specifies the property of the expression).
Segment Word Sequence	Whether a particular sequence of words appears in a segment.	Observable	Nominal	-
Segment Length	The length of a segment in words.	Observable	Ordinal	-
Sub-segment Token	Whether a sub-segment is a particular token (when a sub-segment is a word).	Observable	Nominal	-
Sub-segment POS Tag	Whether a sub-segment corresponds to a particular part-of-speech tag (when a sub-segment is a word).	Observable	Nominal	-

Examples of edges in the segment layer.

Node Pair	Explanation
Segment Type & Segment Observable Characteristics	The observable characteristics of a segment or a sub-segment, such as word sequence, length, token, and POS tag, indicate the information it contains or represents.
Segment Type & Sub-segment Type	The type of a segment indicates the possible types of sub-segments it contains and vice versa. For example, it is more likely to find specific patient demographics, such as age and sex, in a sentence describing the patients involved in a study and vice versa.
Segment Type & Sub-segment Relation Type	The type of a segment indicates the possible types of relations between the sub-segments it contains and vice versa. For example, it is more likely to find a concept-expression pair in which the expression is a representation of the concept in a sentence that defines the concept and vice versa.
Segment Type & Segment Type	A sequence of segments is constructed specifically to express a coherent idea. Therefore, the type of a segment is dependent on those of the other segments in the same sequence.

Examples of edges across layers.

Node Pair	Explanation
Concept Type & Resource Type	The types of resources available for a concept vary by its type. For example, a resource for an operation concept (*e.g.*, Fourier transform) can be a website for a tool package that helps to apply the operation.
Concept Difficulty & Resource Readability	Resources written for more difficult concept are less readable, while concepts commonly described by less readable resources are more difficult.
Concept Genericity & Resource Specificity	Resources written for more generic concepts are less specific, while concepts commonly described by less specific resources are more generic.
Concept Genericity & Resource Type	The types of resources available for a concept vary depending on its genericity. For example, textbooks are commonly written for generic concepts.
Resource Type & Segment Type	The types of segments in a resource vary by its type. For example, it is more likely for exercises to appear in tutorials than in encyclopedia pages.

A.2 Interview Questions[1] for the Math Search System Evaluation

Intro

1. What is your math background? ( ) [O]

   A. Arithmetic only
   B. High school algebra
   C. High school algebra, trigonometry, some calculus
   D. College-level calculus
   E. College-level math beyond calculus

2. Have you searched for information and resources online? ( ) [O]

   A. Yes B. No

3. If the answer to Question 2 is "Yes", how often do you perform such searches? ( ) [O]

   A. A few times per day or more
   B. A few times per week
   C. A few times per month
   D. A few times per year or less.

4. Have you searched for math-related information and resources online? ( ) [O]

   A. Yes B. No

5. If the answer to Question 4 is "Yes", how often do you perform such searches? ( ) [O]

   A. A few times per day or more
   B. A few times per week
   C. A few times per month
   D. A few times per year or less.

[1] Questions followed by the "[O]" symbol are also used in the online version of the evaluation.

For search task __ on search engine A

1. How do you find search task in terms of difficulty? ( ) [o]

A. Very easy B. Easy C. Normal D. Difficult E. Very Difficult

2. If your answer to Question 1 is "Difficult" or "Very Difficult", what difficulties did you encounter in completing the task?

_____
_____
_____

3. Have you done similar tasks before in your everyday life? ( )

A. Yes B. No

4. If your answer to Question 3 is "Yes", what approaches do you take? (Please check all that apply)

- ☐ Perform keyword search using a search engine
- ☐ Visit online math resources (e.g., Wikipedia, MathWorld and ArXiv)
- ☐ Go through offline materials (e.g., hardcopies of textbooks and journals)
- ☐ Ask your teachers/professors/friends
- ☐ Others: _____

5. If your answers for Question 4 include "performing keyword search using a search engine", do you find it cost-effective? Why?

_____
_____
_____

6. If your answers for Question 4 include approaches other than "performing keyword search using a search engine", are those approaches more cost-effective than keyword search? Why?

_____
_____
_____

7. Please briefly describe how you performed the search task using the given search engine.

_____
_____
_____

For feature __

1. When you performed the search task, were you aware that the search engine has feature X? ( ) [O]

A. Yes B. No

2. If your answer to Question 1 is "Yes", did you make use of this feature in completing the search task? ( ) [O]

A. Yes B. No

3. If your answer to Question 2 is "Yes", how did you make use of it? If not, why?

_____
_____
_____

4. If your answer to Question 2 is "Yes", how would you rate this feature for the task in terms of usefulness? ( ) Why? [O]

A. Very unuseful B. Unuseful C. Normal D. Useful E. Very useful

_____
_____
_____

5. If your answer to Question 4 is A-C, how do you think this feature can be improved such that it would help?

_____
_____
_____

For feature __

1. When you performed the search task, were you aware that the search engine has feature X? ( ) [O]

A. Yes B. No

2. If your answer to Question 1 is "Yes", did you make use of this feature in completing the search task? ( ) [O]

A. Yes B. No

3. If your answer to Question 2 is "Yes", how did you make use of it? If not, why?

_____

_____
_____

4. If your answer to Question 2 is "Yes", how would you rate this feature for the task in terms of usefulness? ( ) Why? [O]

A. Very unuseful B. Unuseful C. Normal D. Useful E. Very useful

_____
_____
_____

5. If your answer to Question 4 is A-C, how do you think this feature can be improved such that it would help?

_____
_____
_____

Overall A

1. How would you rate the given search engine for this search task in terms of effectiveness? ( ) [O]

A. Very ineffective B. Ineffective C. Normal D. Effective E. Very effective

2. What (other) functionalities do you think would be useful in completing this search task?

_____
_____
_____

3. Any other comments for the search task or the search engine?

_____
_____
_____

For search task __ on search engine B

1. How do you find this search task in terms of difficulty? ( ) [o]

A. Very easy B. Easy C. Normal D. Difficult E. Very Difficult

2. If your answer to Question 1 is "Difficult" or "Very Difficult", what difficulties did you encounter in completing the task?

_____
_____
_____

3. Have you done similar tasks before in your everyday life? ( )

A. Yes B. No

4. If your answer to Question 3 is "Yes", what approaches do you take? (Please check all that apply)

- ☐ Perform keyword search using a search engine
- ☐ Visit online math resources (e.g., Wikipedia, MathWorld and ArXiv)
- ☐ Go through offline materials (e.g., hardcopies of textbooks and journals)
- ☐ Ask your teachers/professors/friends
- ☐ Others: _____

5. If your answers for Question 4 include "performing keyword search using a search engine", do you find it cost-effective? Why?

_____
_____
_____

6. If your answers for Question 4 include approaches other than "performing keyword search using a search engine", are those approaches more cost-effective than keyword search? Why?

_____
_____
_____

7. Please briefly describe how you performed the search task using the given search engine.

_____
_____
_____

Overall B

1. How would you rate the given search engine for this search task in terms of effectiveness? ( ) [O]

   A. Very ineffective B. Ineffective C. Normal D. Effective E. Very effective

2. What (other) functionalities do you think would be useful in completing this search task?

_____
_____
_____

3. Any other comments for the search task or the search engine?

_____
_____
_____

Final

1. Any comments for this experiment?

_____
_____
_____

A.3 Appreciation Email from the Math Search System Evaluation

Sir,

The search engine you have developed is beautiful, fantastic and so on. It is hardly possible to describe the beauty of your program.i have become a fan of it. I often search for information for two of my kids - one in Class 6 and another in Class 9. I know how difficult and time consuming it is to locate the useful information. I have to find the links, go through each of them to find their contents and then to assess whether it is of the required standard and so on.

I would remain ever grateful if you kindly provide me the link to the search engine software and permit me to use it for my kids.

Hats off to you....

Thanking you in anticipation.....

[Name of sender omitted for privacy concerns.]

A.4 Publications Resulting from this Ph.D Research

Zhao, J., Kan, M.-Y., and Theng, Y. L. (2008). Math information retrieval: User requirements and prototype implementation. In *JCDL '08: Proceedings of the Joint Conference on Digital Libraries*, pages 187–196. ACM Press.

Zhao J. (2010). Towards a user-centric math information retrieval system. *Bulletin of IEEE Technical Committee on Digital Libraries*, 4(2), IEEE Press.

Zhao, J. and Kan, M.-Y. (2010). Domain-specific iterative readability computation. In *JCDL '10: Proceedings of the Joint Conference on Digital Libraries*. ACM Press.

Zhao, J., Kan, M.-Y., Procter, P. M., Zubaidah, S., Yip, W. K., and Li, G. M. (2010). Improving search for evidence-based practice using information extraction. In *AMIA '10: Proceedings of the American Medical Informatics Association Annual Symposium*.

Zhao, J., Kan, M.-Y., Procter, P. M., Zubaidah, S., Yip, W. K., and Li, G. M. (2010). eEvidence: Information seeking support for evidence-based practice: An implementation case study. In *AMIA '10: Proceedings of the American Medical Informatics Association Annual Symposium*.

Zhao, J., Bysani, P., and Kan, M.-Y. (2012). Exploiting classification correlations for the extraction of evidence-based practice information. In *AMIA '12: Proceedings of the American Medical Informatics Association Annual Symposium*.

Bibliography

[Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In *DL '00: Proceedings of the ACM Conference on Digital Libraries*, pages 85–94. ACM Press.

[Aone and Ramos-Santacruz, 2000] Aone, C. and Ramos-Santacruz, M. (2000). REES: A large-scale relation and event extraction system. In *ANLP '00: Proceedings of the Applied Natural Language Processing Conference*, pages 76–83.

[Apps and MacIntyre, 2006] Apps, A. and MacIntyre, R. (2006). Why OpenURL? *D-Lib Magazine*, 12(5).

[Bakken et al., 2008] Bakken, S., Currie, L. M., Lee, N.-J., Roberts, W. D., Collins, S. A., and Cimino, J. J. (2008). Integrating evidence into clinical information systems for nursing decision support. *International Journal of Medical Informatics*, 77(6):413–420.

[Banko et al., 2007] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *IJCAI '07: Proceedings of the International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

[Beit-Arie et al., 2001] Beit-Arie, O., Blake, M., Caplan, P., Flecker, D., Ingoldsby, T., Lannom, L. W., Mischo, W. H., Pentz, E., Rogers, S., and de Sompel, H. V. (2001). Linking to the appropriate copy: Report of a DOI-based prototype. *D-Lib Magazine*, 7(9).

[Bhattacharjya et al., 2009] Bhattacharjya, D., Eidsvik, J., and Mukerji, T. (2009). The value of information in spatial decision making. *Mathematical Geosciences*, 42(2):141–163.

[Bikel et al., 1997] Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: A high-performance learning name-finder. In *ANLC '97: Proceedings of the Conference on Applied Natural Language Processing*, pages 194–201. Association for Computational Linguistics.

[Bishop, 1998] Bishop, A. P. (1998). Digital libraries and knowledge disaggregation: The use of journal article components. In *DL '98: Proceedings of the ACM Conference on Digital Libraries*, pages 29–39. ACM Press.

[Bobic et al., 2012] Bobic, T., Klinger, R., Thomas, P., and Hofmann-Apitius, M. (2012). Improving distantly supervised extraction of drug-drug and protein-protein interactions. In *ROBUS-UNSUP '12: Proceedings of the Joint Workshop on Unsupervised and Semi-supervised Learning in NLP*, pages 35–43. Association for Computational Linguistics.

[Bond, 2005] Bond, C. S. (2005). Nurses and computers: An international perspective on how nurses are, and how they would like to be, using ICT in the workplace, and the support they consider that they need. Technical Report 29, Bournemouth University.

[Borst et al., 2008] Borst, A., Gaudinat, A., Grabar, N., and Boyer, C. (2008). Lexically based distinction of readability levels of health documents. In *MIE '08: Proceedings of the International Congress of the European Federation for Medical Informatics*.

[Bosch et al., 2007] Bosch, A., Muñoz, X., and Martí, R. (2007). Review: Which is the best way to organize/classify images by content? *Image and Vision Computing*, 25(6):778–791.

[Boudin et al., 2010] Boudin, F., Nie, J.-Y., Bartlett, J. C., Grad, R., Pluye, P., and Dawes, M. (2010). Combining classifiers for robust PICO element detection. *BMC Medical Informatics and Decision Making*, 10(1).

[Boutell and Luo, 2005] Boutell, M. and Luo, J. (2005). Beyond pixels: Exploiting camera metadata for photo classification. *Pattern Recognition*, 38(6):935–946.

[Brady and Lewin, 2007] Brady, N. and Lewin, L. (2007). Evidence-based practice in nursing: Bridging the gap between research and practice. *Journal of Pediatric Health Care*, 21(1):53–56.

[Brin, 1999] Brin, S. (1999). Extracting patterns and relations from the World Wide Web. In *WebDB '98: Selected Papers from the International Workshop on the World Wide Web and Databases*, pages 172–183. Springer-Verlag.

[Broder, 2002] Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2):3–10.

[Bruijn et al., 2008] Bruijn, B., Simona, C., Kiritchenko, S., Martin, J., and Sim, I. (2008). Automated information extraction of key trial design elements from clinical trial publications. In *AMIA '08: Proceedings of the American Medical Informatics Association Annual Symposium*.

[Bunescu and Mooney, 2005] Bunescu, R. C. and Mooney, R. J. (2005). A shortest path dependency kernel for relation extraction. In *HLT-EMNLP '05: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics.

[Burstein et al., 2004] Burstein, J., Chodorow, M., and Leacock, C. (2004). Automated essay evaluation: The criterion online writing service. *AI Magazine*, 25(3):27–36.

[Buyko et al., 2012] Buyko, E., Beisswanger, E., and Hahn, U. (2012). The extraction of pharmacogenetic and pharmacogenomic relations – a case study using PharmGKB. In *PSB '12: Proceedings of the Pacific Symposium on Biocomputing*, pages 376–387. Association for Computational Linguistics.

[Carlson et al., 2010] Carlson, A., Betteridge, J., Wang, R. C., Hruschka., E. R., and Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction. In *WSDM '10: Proceedings of the International Conference on Web Search and Web Data Mining*, pages 101–110.

[Carrington et al., 2005] Carrington, P. J., Scott, J., and Wasserman, S. (2005). *Models and Methods in Social Network Analysis (Structural Analysis in the Social Sciences)*. Cambridge University Press.

[Chan and Yeung, 2000] Chan, K.-F. and Yeung, D.-Y. (2000). Mathematical expression recognition: A survey. *International Journal on Document Analysis and Recognition*, 3(1):3–15.

[Chandler et al., 2011] Chandler, A., Wiley, G., and LeBlanc, J. (2011). Towards transparent and scalable OpenURL quality metrics. *D-Lib Magazine*, 17(3-4).

[Chieu and Ng, 2002] Chieu, H. L. and Ng, H. T. (2002). A maximum entropy approach to information extraction from semi-structured and free text. In *AAAI '02: Proceedings of the National Conference on Artificial Intelligence*, pages 786–791.

[Chung, 2009] Chung, G. Y. (2009). Sentence retrieval for abstracts of randomized controlled trials. *BMC Medical Informatics and Decision Making*, 9(1):10.

[Chung and Coiera, 2007] Chung, G. Y. and Coiera, E. (2007). A study of structured clinical abstracts and the semantic classification of sentences. In *BioNLP '07: Proceedings of the Workshop on Biomedical Natural Language Processing*, pages 121–128.

[Ciravegna, 2001] Ciravegna, F. (2001). Adaptive information extraction from text by rule induction and generalisation. In *IJCAI '01: Proceedings of the*

International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc.

[Cohen, 1960] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

[Coleman and Liau, 1975] Coleman, M. and Liau, T. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284.

[Collins-Thompson and Callan, 2004] Collins-Thompson, K. and Callan, J. P. (2004). A language modeling approach to predicting reading difficulty. In *HLT-NAACL '04: Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 193–200. Association for Computational Linguistics.

[Crestani et al., 2003] Crestani, F., de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2003). A multi-layered Bayesian network model for structured document retrieval. In *ECSQARU '03: Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 74–86. Springer-Verlag.

[Culotta and Sorensen, 2004] Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *ACL '04: Proceedings of the Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

[Cunningham et al., 2002] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL '02: Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 168–175. Association for Computational Linguistics.

[Dale and Chall, 1948] Dale, E. and Chall, J. S. (1948). A formula for predicting readability: Instructions. *Educational Research Bulletin*, pages 37–54.

[Dalvi et al., 2012] Dalvi, B. B., Cohen, W. W., and Callan, J. (2012). Websets: Extracting sets of entities from the web using unsupervised information extraction. In *WSDM '12: Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 243–252. ACM Press.

[de Campos et al., 2000] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2000). Building Bayesian network-based information retrieval systems. In *DEXA '00: Proceedings of the International Workshop on Database and Expert Systems Applications*, pages 543–550. IEEE Press.

[de Campos et al., 2004] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2004). Clustering terms in the Bayesian network retrieval model: A new approach with two term-layers. *Applied Soft Computing*, 4(2):149–158.

[de Campos et al., 2006] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2006). Retrieving medical records using Bayesian networks. *Encyclopaedia of Data Warehousing and Mining*, pages 960–964.

[de Campos et al., 2008] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2008). An information retrieval system for parliamentary documents. *Bayesian Networks: A Practical Guide to Applications*, pages 203–223.

[Demner-Fushman and Lin, 2007] Demner-Fushman, D. and Lin, J. (2007). Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 1(33):63–103.

[Demner-Fushman et al., 2008] Demner-Fushman, D., Seckman, C., Fisher, C., Hauser, S. E., Clayton, J., and Thoma, G. R. (2008). A prototype system to support evidence-based practice. In *AMIA '08: Proceedings of the American Medical Informatics Association Annual Symposium*.

[Descombes et al., 1998] Descombes, X., Kruggel, F., and von Cramon, D. Y. (1998). Spatio-temporal fMRI analysis using Markov random fields. *Transactions on Medical Imaging*, 17(6):1028–1039.

[Doddington et al., 2004] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ACE) program: Tasks, data, and evaluation. In *LREC '04: Proceedings of the International Conference on Language Resources and Evaluation*, pages 837–840.

[DuBay, 1990] DuBay, W. H. (1990). *Unlocking Language: The Classic Readability Studies*. BookSurge Publishing.

[Eisenberg and Berkowitz, 1990] Eisenberg, M. B. and Berkowitz, R. E. (1990). *Information Problem-solving: The Big Six Skills Approach to Library and Information Skills Instruction*. Albex Publishing.

[Entity Linking, 2011] Entity Linking (2011). Proposed task description for knowledge-based population at TAC 2011. *Population English Edition*, pages 1–13.

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

[Feng and Manmatha, 2008] Feng, S. and Manmatha, R. (2008). A discrete direct retrieval model for image and video retrieval. In *CIVR '08: Proceedings of the International Conference on Content-based Image and Video Retrieval*, pages 427–436. ACM Press.

[Feng et al., 2003] Feng, Y., Zhuang, Y., and Pan, Y. (2003). Music information retrieval by detecting mood via computational media aesthetics. In *WI '03: Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 235–241. IEEE Press.

[Fineout-Overholt et al., 2005] Fineout-Overholt, E., Melnyk, B. M., and Schultz, A. (2005). Transforming health care from the inside out: Advancing evidence-based practice in the 21st century. *Journal of Professional Nursing*, 21(6):335–344.

[Fleiss, 1971] Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

[Flesch, 1948] Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

[Foote et al., 2002] Foote, J., Cooper, M., and Nam, U. (2002). Audio retrieval by rhythmic similarity. In *Proceedings of the International Conference on Music Information Retrieval*, pages 265–266.

[Friedman et al., 2000] Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. In *RECOMB '00: Proceedings of the Annual International Conference on Computational Molecular Biology*, pages 127–135.

[Fung and Favero, 1995] Fung, R. and Favero, B. D. (1995). Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–57.

[Gliozzo et al., 2005] Gliozzo, A. M., Giuliano, C., and Rinaldi, R. (2005). Instance filtering for entity recognition. *SIGKDD Explorations Newsletter*, 7(1):11–18.

[Graber et al., 1999] Graber, M. A., Roller, C. M., and Kaeble, B. (1999). Readability levels of patient education material on the World Wide Web. *Journal of Family Practice*, 48(1):58–61.

[Gray and Leary, 1935] Gray, W. S. and Leary, B. (1935). *What Makes a Book Readable*. Chicago Press.

[GuoDong et al., 2005] GuoDong, Z., Jian, S., Jie, Z., and Min, Z. (2005). Exploring various knowledge in relation extraction. In *ACL '05: Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics.

[Hakenberg et al., 2008] Hakenberg, J., Plake, C., Royer, L., Strobelt, H., Leser, U., and Schroeder, M. (2008). Gene mention normalization and interaction extraction with context models and sentence motifs. *Genome Biology*, 9(Suppl 2).

[Heilman et al., 2007] Heilman, M., Collins-Thompson, K., Callan, J., and Eskenazi, M. (2007). Combining lexical and grammatical features to improve readability measures for first and second language texts. In *HLT-NAACL '07: Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting*, pages 460–467. Association for Computational Linguistics.

[Hliaoutakis et al., 2006] Hliaoutakis, A., Varelas, G., Petrakis, E. G. M., and Milios, E. (2006). Medsearch: A retrieval system for medical information based on semantic similarity. In *ECDL '06: Proceedings of the European Conference on Digital Libraries*, pages 512–515.

[Hobbs et al., 1997] Hobbs, J. R., Appelt, D. E., Bear, J., Israel, D. J., Kameyama, M., Stickel, M. E., and Tyson, M. (1997). FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. *Finite-State Language Processing*, pages 383–406.

[Isozaki and Kazawa, 2002] Isozaki, H. and Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *COLING '02: Proceedings of the International Conference on Computational Linguistics*, pages 1–7. Association for Computational Linguistics.

[Jayram et al., 2006] Jayram, T. S., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Zhu, H. (2006). Avatar information extraction system. *Data Engineering Bulletin*, 29(1):40–48.

[Jensen and Nielsen, 2007] Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer-Verlag, 2nd edition.

[Jiang and Zhai, 2007] Jiang, J. and Zhai, C. (2007). A systematic exploration of the feature space for relation extraction. In *HLT-ACL '07: Proceedings of Human Language Technologies and the Annual Meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics.

[Jiang et al., 2004] Jiang, S., Huang, T., and Gao, W. (2004). An ontology-based approach to retrieve digitized art images. In *WI '04: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 131–137. IEEE Press.

[Junker and Schreiber, 2008] Junker, B. H. and Schreiber, F. (2008). *Analysis of Biological Networks*. Wiley-Interscience.

[Kambhatla, 2004] Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *ACLdemo '04: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics.

[Kim et al., 2007] Kim, H., Goryachev, S., Rosemblat, G., Browne, A., Keselman, A., and Zeng-Treitler, Q. (2007). Beyond surface characteristics: A new health text-specific readability measurement. In *AMIA '07: Proceedings of the American Medical Informatics Association Annual Symposium*.

[Kim et al., 2011] Kim, J.-D., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N., and Tsujii, J. (2011). Overview of BioNLP shared task 2011. In *BioNLP-ST '11: Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6. Association for Computational Linguistics.

[Kim and Compton, 2001] Kim, M. and Compton, P. (2001). Formal concept analysis for domain-specific document retrieval systems. In *AI '01: Proceedings of the Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, pages 237–248. Springer-Verlag.

[Kim et al., 2010] Kim, S. N., Martinez, D., and Cavedon, L. (2010). Automatic classification of sentences for evidence based medicine. In *DTMBIO '10: Proceedings of the ACM International Workshop on Data and Text Mining in Biomedical Informatics*, pages 13–22. ACM Press.

[Kindermann and Snell, 1980] Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. American Mathematical Society.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

[Kohlhase et al., 2012] Kohlhase, M., Matican, B. A., and Prodescu, C. C. (2012). Mathwebsearch 0.5 – scaling an open formula search engine. In *CICM '12: Proceedings of the Conferences on Intelligent Computer Mathematics*.

[Kohlhase and Sucan, 2006] Kohlhase, M. and Sucan, I. (2006). A search engine for mathematical formulae. In *AISC '06: Proceedings of Artificial Intelligence and Symbolic Computation*, pages 241–253. Springer-Verlag.

[Krallinger et al., 2011] Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., Chatr-aryamontri, A., Winter, A., Perfetto, L., Briganti, L., Licata, L., Iannuccelli, M., Castagnoli, L., Cesareni, G., Tyers, M., Schneider, G., Rinaldi, F., Leaman, R., Gonzalez, G., Matos, S., Kim, S., Wilbur, W., Rocha, L., Shatkay, H., Tendulkar, A., Agarwal, S., Liu, F., Wang, X., Rak, R., Noto, K., Elkan, C., and Lu, Z. (2011). The protein-protein interaction tasks of BioCreative III: Classification/Ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics*, 12(Suppl 8):S3.

[Krishnamurthy et al., 2008] Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F., Vaithyanathan, S., and Zhu, H. (2008). SystemT: A system for declarative information extraction. *SIGMOD Record*, 37(4):7–13.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and

labeling sequence data. In *ICML '01: Proceedings of the International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc.

[Lang et al., 2010] Lang, H., Metzler, D., Wang, B., and Li, J.-T. (2010). Improved latent concept expansion using hierarchical Markov random fields. In *CIKM'10: Proceedings of the ACM Conference of Information and Knowledge Management*, pages 249–258. ACM Press.

[Lay and Florio, 1996] Lay, P. and Florio, T. (1996). The use of readability formulas in health care. *Psychology Health Medicine*, 1(1):7–28.

[Lease, 2009] Lease, M. (2009). An improved Markov random field model for supporting verbose queries. In *SIGIR'09: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 476–483. ACM Press.

[Lee and Myaeng, 2002] Lee, Y.-B. and Myaeng, S. H. (2002). Text genre classification with genre-revealing and subject-revealing features. In *SIGIR '02: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 145–150. ACM Press.

[Lempel and Moran, 2000] Lempel, R. and Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1-6):387–401.

[Leroy et al., 2008] Leroy, G., Miller, T., Rosemblat, G., and Browne, A. (2008). A balanced approach to health information evaluation: A vocabulary-based naïve-Bayes classifier and readability formulas. *Journal of the American Society for Information Science and Technology*.

[Liu et al., 2010] Liu, B., Chiticariu, L., Chu, V., Jagadish, H. V., and Reiss, F. (2010). Automatic rule refinement for information extraction. *Proceedings of the VLDB Endowment*, 3(1-2):588–597.

[Liu et al., 2008] Liu, M., Liu, Y., Xiang, L., Chen, X., and Yang, Q. (2008). Extracting key entities and significant events from online daily news. In *IDEAL '08: Proceedings of the Intelligent Data Engineering and Automated Learning*, pages 201–209. Springer-Verlag.

[Lively and Pressey, 1923] Lively, B. A. and Pressey, S. L. (1923). A method for measuring the 'vocabulary burden' of textbooks. *Educational Administration and Supervision*, 9(5):389–398.

[Llorente et al., 2010] Llorente, A., Manmatha, R., and Rüger, S. (2010). Image retrieval using Markov random fields and global image features. In *CIVR '10: Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 243–250. ACM Press.

[Logan and Salomon, 2001] Logan, B. and Salomon, A. (2001). A music similarity function based on signal analysis. In *ICME '01: Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE Press.

[Mani et al., 2005] Mani, S., Valtorta, M., and McDermott, S. (2005). Building Bayesian network models in medicine: The mentor experience. *Applied Intelligence*, 22(2):93–108.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[McCallum, 2006] McCallum, A. (2006). Information extraction, data mining and joint inference. In *KDD '06: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 835–835. ACM Press.

[McCallum et al., 2000] McCallum, A., Freitag, D., and Pereira, F. C. N. (2000). Maximum entropy Markov models for information extraction and segmentation. In *ICML '00: Proceedings of the International Conference on Machine Learning*, pages 591–598. Morgan Kaufmann Publishers Inc.

[McDonald et al., 2005] McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y., and White, P. (2005). Simple algorithms for complex relation extraction with applications to biomedical IE. In *ACL '05: Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 491–498. Association for Computational Linguistics.

[McKinney and Breebaart, 2003] McKinney, M. F. and Breebaart, J. (2003). Features for audio and music classification. In *ISMIR '03: Proceedings of the International Conference on Music Information Retrieval*.

[McLaughlin, 1969] McLaughlin, H. G. (1969). SMOG grading - a new readability formula. *Journal of Reading*, pages 639–646.

[Mcnamara et al., 2002] Mcnamara, D. S., Louwerse, M. M., and Graesser, A. C. (2002). Coh-Metrix: Automated cohesion and coherence scores to predict text readability and facilitate comprehension. Technical report, University of Memphis.

[Meij et al., 2009] Meij, E., Trieschnigg, D., de Rijke, M., and Kraaij, W. (2009). Conceptual language models for domain-specific retrieval. *Information Processing & Management*, 46(4):448–469.

[Melnyk and Fineout-Overholt, 2000] Melnyk, B. and Fineout-Overholt, E. (2000). *Evidence-based Practice in Nursing and Healthcare (2nd Edition)*. Wolters Kluwer/Lippincott Williams and Wilkins.

[Merlin and Persson, 1996] Merlin, G. and Persson, O. (1996). Studying research collaboration using co-authorships. *Scientometrics*, 36(3):363–377.

[Metzler and Croft, 2004] Metzler, D. and Croft, W. B. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750.

[Metzler and Croft, 2005] Metzler, D. and Croft, W. B. (2005). A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479. ACM Press.

[Metzler and Croft, 2007] Metzler, D. and Croft, W. B. (2007). Latent concept expansion using Markov random fields. In *SIGIR '07: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–318. ACM Press.

[Michelakis et al., 2009] Michelakis, E., Krishnamurthy, R., Haas, P. J., and Vaithyanathan, S. (2009). Uncertainty management in rule-based information extraction systems. In *SIGMOD '09: Proceedings of the SIGMOD International Conference on Management of Data*, pages 101–114. ACM Press.

[Miner and Munavalli, 2007] Miner, R. and Munavalli, R. (2007). An approach to mathematical search through query formulation and data normalization. In *MKM '07: Towards Mechanized Mathematical Assistants, Proceedings of the International Conference on Mathematical Knowledge Management*, pages 342–355.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 1003–1011. Association for Computational Linguistics.

[Mitra et al., 2007] Mitra, P., Giles, C. L., Sun, B., and Liu, Y. (2007). Chemxseer: A digital library and data repository for chemical kinetics. In *CIMS '07: Proceedings of the ACM Workshop on CyberInfrastructure: Information Management in eScience*, pages 7–10. ACM Press.

[Muslea, 1999] Muslea, I. (1999). Extraction patterns for information extraction tasks: A survey. In *AAAI '99: Proceedings of the National Conference on Artificial Intelligence: Workshop on Machine Learning for Information Extraction*. AAAI Press.

[Nadeau, 2007] Nadeau, D. (2007). *Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision*. PhD thesis, University of Ottawa.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.

[National Health and Medical Research Council, 1999] National Health and Medical Research Council (1999). *NHMRC: A Guide to the Development, Implementation and Evaluation of Clinical Practice Guidelines.* National Health and Medical Research Council.

[Oremann, 2007] Oremann, M. H. (2007). Internet resources for evidence-based practice in nursing. *Plastic Surgical Nursing*, 27(1):37–39.

[Page et al., 1998] Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

[Pearl, 1985] Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *CogSci'85: Proceedings of the Conference of the Cognitive Science Society*, pages 329–334.

[Piskorski et al., 2008] Piskorski, J., Tanev, H., Atkinson, M., and Van Der Goot, E. (2008). Cluster-centric approach to news event extraction. In *Proceedings of the Conference on New Trends in Multimedia and Network Information Systems*, pages 276–290. IOS Press.

[Piskorski et al., 2007] Piskorski, J., Tanev, H., and Wennerberg, P. O. (2007). Extracting violent events from on-line news for ontology population. In *BIS '07: Proceedings of the International Conference on Business Information Systems*, pages 287–300. Springer-Verlag.

[Pitler and Nenkova, 2008] Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In *EMNLP'08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195.

[Pohle et al., 2006] Pohle, T., Knees, P., Schedl, M., and Widmer, G. (2006). Independent component analysis for music similarity computation. In *ISMIR '06: Proceedings of the International Society for Music Information Retrieval*, pages 228–233.

[Poon and Domingos, 2007] Poon, H. and Domingos, P. (2007). Joint inference in information extraction. In *AAAI'07: Proceedings of the National Conference on Artificial Intelligence*, pages 913–918. AAAI Press.

[Price et al., 2007] Price, S. L., Nielsen, M. L., Delcambre, L. M. L., and Vedsted, P. (2007). Semantic components enhance retrieval of domain-specific documents. In *CIKM'07: Proceedings of the ACM Conference of Information and Knowledge Management*, pages 429–438. ACM Press.

[Price et al., 2009] Price, S. L., Nielsen, M. L., Delcambre, L. M. L., Vedsted, P., and Steinhauer, J. (2009). Using semantic components to search for domain-specific documents: An evaluation from the system perspective and the user perspective. *Information System*, 34(8):724–752.

[Quellec et al., 2008] Quellec, G., Lamard, M., Bekri, L., Cazuguel, G., Roux, C., and Cochener, B. (2008). Multimodal medical case retrieval using Bayesian networks and the Dezert-Smarandache theory. In *ISBI'08: Proceedings of the IEEE International Symposium on Biomedical Imaging*, pages 245–248. IEEE Press.

[Quellec et al., 2011] Quellec, G., Lamard, M., Cazuguel, G., Roux, C., and Cochener, B. (2011). Case retrieval in medical databases by fusing heterogeneous information. *Transactions on Medical Imaging*, 30(1):108–118.

[Radhouani et al., 2009] Radhouani, S., Jiang, C.-L. M., and Falquet, G. (2009). FlexIR: A domain-specific information retrieval system. *Polibits*, 39(27-31):2.

[Reiss et al., 2008] Reiss, F., Raghavan, S., Krishnamurthy, R., Zhu, H., and Vaithyanathan, S. (2008). An algebraic approach to rule-based information extraction. In *ICDE'08: Proceedings of the IEEE International Conference on Data Engineering*, pages 933–942. IEEE Press.

[Reuters, 2012] Reuters, T. (2012). The Thomson Reuters impact factor. http://thomsonreuters.com/products_services/science/free/essays/impact_factor/. [Online: Accessed 5-July-2012].

[Riedel and McCallum, 2011] Riedel, S. and McCallum, A. (2011). Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *BioNLP-ST'11: Proceedings of BioNLP Shared Task 2011 Workshop*, pages 46–50. Association for Computational Linguistics.

[Roth and Yih, 2001] Roth, D. and Yih, W. (2001). Relational learning via propositional algorithms: An information extraction case study. In *IJCAI'01: Proceedings of the International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

[Sackett et al., 2000] Sackett, D. L., Straus, S. E., Richardson, W. S., Rosenberg, W., and Haynes, R. B. (2000). *Evidence-based Medicine: How to Practice and Teach EBM*. Churchill Livingstone, 2nd edition.

[Sarawagi, 2008] Sarawagi, S. (2008). Information extraction. *Foundations and Trends in Databases*, 1(3):261–377.

[Scaringella, 2008] Scaringella, N. (2008). Timbre and rhythmic trap-tandem features for music information retrieval. In *ISMIR'08: Proceedings of the International Society for Music Information Retrieval*, pages 626–631.

[Scaringella et al., 2006] Scaringella, N., Zoia, G., and Mlynek, D. (2006). Automatic genre classification of music content: A survey. *Signal Processing Magazine*, 23(2):133–141.

[Schapke and Scherer, 2004] Schapke, S.-E. and Scherer, R. J. (2004). A four-layer Bayesian network for product model based information mining. In *ICCCBE'04: Proceedings of the International Conference on Computing in Civil and Building Engineering*.

[Schlüter and Osendorfer, 2011] Schlüter, J. and Osendorfer, C. (2011). Music similarity estimation with the mean-covariance restricted Boltzmann machine. In *ICMLA'11: Proceedings of the International Conference on Machine Learning and Applications*.

[Schuller et al., 2003] Schuller, B., Zobl, M., Rigoll, G., and Lang, M. (2003). A hybrid music retrieval system using belief networks to integrate multimodal queries and contextual knowledge. In *ICME'03: Proceedings of the International Conference on Multimedia and Expo*, pages 57–60. IEEE Press.

[Schwarm and Ostendorf, 2005] Schwarm, S. E. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *ACL'05: Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Survey*, 34(1):1–47.

[Seymore et al., 1999] Seymore, K., Mccallum, A., and Rosenfeld, R. (1999). Learning hidden Markov model structure for information extraction. In *AAAI'99: Proceedings of the National Conference on Artificial Intelligence: Workshop on Machine Learning for Information Extraction*, pages 37–42.

[Shen et al., 2007] Shen, W., Shen, A., Naughton, J. F., and Ramakrishnan, R. (2007). Declarative information extraction using datalog with embedded extraction predicates. In *VLDB'07: Proceedings of the International Conference on Very Large Databases*, pages 1033–1044. VLDB Endowment.

[Silveira and Ribeiro-Neto, 2004] Silveira, M. L. and Ribeiro-Neto, B. (2004). Concept-based ranking: A case study in the juridical domain. *Information Processing and Management*, 40(5):791–805.

[Sim et al., 2001] Sim, I., Gorman, P., Greenes, R. A., Haynes, R. B., Kaplan, B., Lehmann, H., and Tang, P. C. (2001). Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 8(6):527–534.

[Sitter and Daelemans, 2003] Sitter, d. and Daelemans, W. (2003). Information extraction via double classification. In *ATEM '03: Proceedings of the International Workshop on Adaptive Text Extraction and Mining*, pages 66–73.

[Skounakis et al., 2003] Skounakis, M., Craven, M., and Ray, S. (2003). Hierarchical hidden Markov models for information extraction. In *IJCAI '03, Proceedings of the International Joint Conference on Artificial Intelligence*, pages 427–433.

[Smith and Senter, 1967] Smith, E. A. and Senter, R. J. (1967). Automated readability index. Technical report, University of Cincinnati.

[Soderland, 1999] Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272.

[Sojka and Líška, 2011] Sojka, P. and Líška, M. (2011). The art of mathematics retrieval. In *DocEng '11: Proceedings of the ACM Symposium on Document Engineering*, pages 57–60. ACM Press.

[Spearman, 1987] Spearman, C. (1987). The proof and measurement of association between two things. *The American Journal of psychology*, 100(3-4):441–471.

[Steyvers and Griffiths, 2007] Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning*.

[Suchanek et al., 2007] Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *WWW '07: Proceedings of the International Conference on World Wide Web*, pages 697–706. ACM Press.

[Tanabe and Wilbur, 2002] Tanabe, L. and Wilbur, W. J. (2002). Tagging gene and protein names in full text articles. In *BioMed '02: Proceedings of the Annual Meeting of Computational Linguistics: Workshop on Natural Language Processing in the Biomedical Domain*, pages 9–13. Association for Computational Linguistics.

[Thelwall, 2004] Thelwall, M. (2004). *Link Analysis: An Information Science Approach*. Academic Press.

[Thorndike, 1921] Thorndike, E. L. (1921). *The Teacher's Word Book*. Teacher's College, Bureau of Publication, Columnbia University, New York City.

[Tikk et al., 2010] Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., and Leser, U. (2010). A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature. *PLoS computational biology*, 6(7).

[Tsikrika and Lalmas, 2004] Tsikrika, T. and Lalmas, M. (2004). Combining evidence for web retrieval using the inference network model: An experimental study. *Information Processing and Management*, 40(5):751–772.

[Turtle and Croft, 1991] Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *Transactions on Information System*, 9(3):187–222.

[Vogel and Washburne, 1928] Vogel, M. and Washburne, C. (1928). An objective method of determining grade placement of children's reading material. *The Elementary School Journal*, 28(5):373–381.

[Wei and Li, 2007] Wei, Z. and Li, H. (2007). A Markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537–1544.

[Wetzler et al., 2009] Wetzler, P. G., Bethard, S., Butcher, K., Martin, J. H., and Sumner, T. (2009). Automatically assessing resource quality for educational digital libraries. In *WICOW '09: Proceedings of the Workshop on Information Credibility on the Web*, pages 3–10. ACM Press.

[Wikipedia, 2012a] Wikipedia (2012a). OpenURL knowledge base — Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/OpenURL_knowledge_base. [Online: Accessed 27-June-2012].

[Wikipedia, 2012b] Wikipedia (2012b). Power iteration — Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Power_iteration. [Online: Accessed 6-June-2013].

[Witten et al., 1999] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *DL '04: Proceedings of the ACM Conference on Digital Libraries*, pages 254–255. ACM Press.

[Wong et al., 2008] Wong, T.-L., Lam, W., and Wong, T.-S. (2008). An unsupervised framework for extracting and normalizing product attributes from multiple web sites. In *SIGIR '08: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42. ACM Press.

[Yan et al., 2006] Yan, X., Song, D., and Li, X. (2006). Concept-based document readability in domain specific information retrieval. In *CIKM'06: Proceedings of the ACM Conference of Information and Knowledge Management*, pages 540–549. ACM Press.

[Yang and Pedersen, 1997] Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML'97: Proceedings of the International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers Inc.

[Yu et al., 2009] Yu, M., Wang, M., Zuo, J., and Zou, X. (2009). Transferring Markov network for information retrieval. In *JCAI '09: Proceedings of the International Joint Conference on Artificial Intelligence*, pages 567–571. IEEE Press.

[Zelenko et al., 2002] Zelenko, D., Aone, C., and Richardella, A. (2002). Kernel methods for relation extraction. In *EMNLP '02: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 71–78. Association for Computational Linguistics.

[Zhang et al., 2006] Zhang, M., Zhang, J., Su, J., and Zhou, G. (2006). A composite kernel to extract relations between entities with both flat and structured features. In *COLING-ACL '06: Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics.

[Zhao and Kan, 2010] Zhao, J. and Kan, M.-Y. (2010). Domain-specific iterative readability computation. In *JCDL '10: Proceedings of the Joint Conference on Digital Libraries*. ACM Press.

[Zhao et al., 2010] Zhao, J., Kan, M.-Y., Procter, P. M., Zubaidah, S., Yip, W. K., and Li, G. M. (2010). Improving search for evidence-based practice using information extraction. In *AMIA '10: Proceedings of the American Medical Informatics Association Annual Symposium*.

[Zhao et al., 2008] Zhao, J., Kan, M.-Y., and Theng, Y. L. (2008). Math information retrieval: User requirements and prototype implementation. In *JCDL '08: Proceedings of the Joint Conference on Digital Libraries*, pages 187–196. ACM Press.

[Zhou et al., 2010] Zhou, G., Qian, L., and Fan, J. (2010). Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Science*, 180(8):1313–1325.

[Zhou et al., 2007] Zhou, G., Zhang, M., Ji, D.-H., and Zhu, Q. (2007). Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *EMNLP-CoNLL '07: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 728–736. Association for Computational Linguistics.

[Zhu et al., 2001] Zhu, Y., Kankanhalli, M. S., and Xu, C. (2001). Pitch tracking and melody slope matching for song retrieval. In *PCM '01: Proceedings of the IEEE Pacific Rim Conference on Multimedia*, pages 530–537. Springer-Verlag.

[Zirnhelt and Breckon, 2007] Zirnhelt, S. and Breckon, T. (2007). Artwork image retrieval using weighted colour and texture similarity. In *Proceedings of the European Conference on Visual Media Production*, pages 2–8. IET Press.